



# 目次

---

- 1. 改訂情報
- 2. ViewCreator について
  - 2.1. 対応するフィールドの型
- 3. 基本設定
  - 3.1. ViewCreator全体の権限設定
  - 3.2. クエリの作成
  - 3.3. クエリビルダによるクエリの作成
  - 3.4. SQLビルダによるクエリの作成
  - 3.5. データ参照の作成
- 4. 応用設定
  - 4.1. 抽出条件の設定
  - 4.2. 外部ページへの連携（リスト/サマリ集計のみ）
  - 4.3. ポータルとの連携
  - 4.4. 物理Viewの作成と表示
  - 4.5. 国際化とヘッダ・フッタ
  - 4.6. 日付フォーマット
  - 4.7. インポート・エクスポート
  - 4.8. テーブル名とフィールド名のキャプション表示
  - 4.9. 外部データソースの利用
  - 4.10. 計算式カラムの設定
  - 4.11. リスト集計におけるCSV出力の設定
  - 4.12. CSV出力ジョブの登録
  - 4.13. ソースコード出力
  - 4.14. 帳票出力の設定
  - 4.15. null値に対する表示文字列の設定
  - 4.16. グラフ集計で「凡例」の表示方法を変更する
  - 4.17. クエリ編集画面でテーブル単位のアクセス権を設定する
  - 4.18. REST APIを参照する
  - 4.19. ルーティングの作成

変更年月日	変更内容
2012-10-01	初版
2013-04-01	第2版 下記を追加・変更しました <ul style="list-style-type: none"> <li>▪ 「<a href="#">対応するフィールドの型</a>」を追加</li> <li>▪ 「<a href="#">外部データソースの利用</a>」を追加</li> </ul>
2013-07-01	第3版 下記を追加・変更しました <ul style="list-style-type: none"> <li>▪ 「<a href="#">リソースの設定項目</a>」を追加</li> <li>▪ 「<a href="#">キー名の変更</a>」を追加</li> <li>▪ 「<a href="#">計算式カラムの設定</a>」を追加</li> <li>▪ 「<a href="#">リスト集計におけるCSV出力の設定</a>」を追加</li> </ul>
2013-10-01	第4版 下記を追加・変更しました <ul style="list-style-type: none"> <li>▪ 「<a href="#">対応するフィールドの型</a>」を追加</li> <li>▪ 「<a href="#">サマリ集計の設定</a>」を追加</li> <li>▪ 「<a href="#">表示設定</a>」を追加</li> <li>▪ 「<a href="#">リスト集計からグラフ集計を作成</a>」を追加</li> </ul>
2014-01-01	第5版 下記を追加・変更しました <ul style="list-style-type: none"> <li>▪ 「<a href="#">アイコン画像の表示</a>」を追加</li> <li>▪ 「<a href="#">検索設定について</a>」に設定可能な動的パラメータについての記載を追加</li> <li>▪ 「<a href="#">外部ページへの連携（リスト/サマリ集計のみ）</a>」にカラム単位でのURL設定についての記載を追加</li> <li>▪ 「<a href="#">外部ページへの連携（リスト/サマリ集計のみ）</a>」にターゲットウィンドウについての記載を追加</li> <li>▪ 「<a href="#">外部ページへの連携（リスト/サマリ集計のみ）</a>」にPathVariablesについての記載を追加</li> <li>▪ 「<a href="#">国際化とヘッダ・フッタ</a>」にカラムのラベルを国際化対応についての記載を追加</li> <li>▪ 「<a href="#">インポート・エクスポート</a>」にインポート・エクスポートに関する注意事項を追加</li> <li>▪ 「<a href="#">クエリの作成</a>」にインポート・エクスポートに関する注意事項を追加</li> </ul>
2014-04-01	第6版 下記を追加・変更しました <ul style="list-style-type: none"> <li>▪ 「<a href="#">利用可能な関数一覧</a>」にFORMAT関数へのロケールの指定についての記載を追加</li> <li>▪ 「<a href="#">参照権の設定</a>」にゲストユーザと認証済みユーザへのデータ参照権についての記載を追加</li> <li>▪ 「<a href="#">国際化とヘッダ・フッタ</a>」にヘッダとフッタのリッチテキストについての記載を追加</li> <li>▪ 「<a href="#">計算式カラムの設定</a>」に計算式入力ダイアログについての記載を追加</li> <li>▪ 「<a href="#">作成したデータ参照の閲覧</a>」にCSVファイルをMicrosoft Excelで読み込む場合についてのコラムを追加</li> <li>▪ 「<a href="#">動的パラメータ</a>」にリクエストパラメータを動的な抽出条件や検索設定の条件値として受け取る「&lt;%REQUEST_PARAMETER%&gt;タグ」を追加</li> </ul>
2014-08-01	第7版 下記を追加・変更しました <ul style="list-style-type: none"> <li>▪ 新しいデータ型（真偽値型）を対応</li> <li>▪ 「<a href="#">複数の表示条件設定（ヒートマップの作成）</a>」を追加</li> <li>▪ 「<a href="#">CSV出力ジョブの登録</a>」を追加</li> <li>▪ 「<a href="#">ソースコード出力</a>」を追加</li> <li>▪ 「<a href="#">帳票出力の設定</a>」を追加</li> <li>▪ 「<a href="#">null値に対する表示文字列の設定</a>」を追加</li> <li>▪ 「<a href="#">応用設定</a>」 - 「<a href="#">グラフ集計用モジュールの切り替え</a>」を追加</li> <li>▪ 「<a href="#">利用可能な関数一覧</a>」に関数を追加</li> </ul>

変更年月日	変更内容
2014-12-01	<p>第8版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ intra-mart Accel Platform 2014 Spring(Granada) で追加、改善された機能に利用可能なバージョンについての記載を追加</li> <li>▪ 「<a href="#">グラフ集計の作成</a>」にグラフ集計の「目盛りの最大値」、「目盛りの最小値」、「1目盛りあたりの数」の自動計算機能を追加</li> <li>▪ 「<a href="#">リスト集計におけるCSV出力の設定</a>」に設定可能項目を追加</li> <li>▪ 「検索の条件値入力にコンボボックスを利用する」を追加</li> <li>▪ 「<a href="#">表示設定</a>」に整列で縦方向の表示位置を設定可能になった旨の記載を追加</li> <li>▪ 「<a href="#">グラフ集計で「凡例」の表示方法を変更する</a>」を追加</li> <li>▪ 「<a href="#">クエリ編集画面でテーブル単位のアクセス権を設定する</a>」を追加</li> </ul>
2015-04-01	<p>第9版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">インポート・エクスポート</a>」に IM-FormaDesigner for Accel Platform についての注意事項を追加</li> <li>▪ 「<a href="#">計算式カラムの設定</a>」に関数の戻り値の説明や、利用方法についての具体例を追加</li> </ul>
2015-08-01	<p>第10版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">ViewCreator全体の権限設定</a>」に認可リソース「管理者」についての記載を追加</li> </ul>
2015-12-01	<p>第11版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">クロス集計</a>」の画面インタフェースと機能を一新</li> <li>▪ 「<a href="#">応用設定</a>」 - 「グラフ集計用モジュールの切り替え」 - 「Highcharts連携モジュールの利用」を追加</li> <li>▪ 「<a href="#">作成したデータ参照の閲覧</a>」に「CSVファイルのヘッダ文字列」に関するコラムを追加</li> <li>▪ 「<a href="#">リスト集計の作成</a>」のソート順に関するコラムに記載を追加</li> </ul>
2016-04-01	<p>第12版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">リスト集計におけるCSV出力の設定</a>」にBOMに付与設定についての記載を追加</li> <li>▪ 「<a href="#">動的パラメータ</a>」の&lt;%REQUEST_PARAMETER(KEY_NAME)%&gt;の説明、および、使用例に関するコラムに記載を追加</li> <li>▪ 「<a href="#">検索設定について</a>」にクロス集計で検索設定が利用可能になった旨の記載を追加</li> <li>▪ 「<a href="#">ソースコード出力</a>」に出力されたソースコードの利用手順を追加</li> </ul>
2016-08-01	<p>第13版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">計算式カラムの設定</a>」の注意事項に計算式が実行されない場合についての記載を追加</li> <li>▪ 「<a href="#">グラフ集計の作成</a>」 - 「<a href="#">JFreeChart</a>」を追加</li> <li>▪ 「<a href="#">グラフ集計の作成</a>」 - 「<a href="#">Highcharts</a>」を追加</li> <li>▪ 「<a href="#">応用設定</a>」 - 「グラフ集計用モジュールの切り替え」を削除</li> </ul>
2016-12-01	<p>第14版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">対応するフィールドの型</a>」にSAP HANAに関する記載を追加、DB2に関する記載を削除</li> <li>▪ 「<a href="#">クロス集計</a>」にクロス集計のカラム設定に注意事項を追加</li> </ul>
2017-08-01	<p>第15版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">クエリの作成</a>」の説明を修正</li> <li>▪ 「<a href="#">クエリの作成</a>」に「<a href="#">クエリの作成 (2017 Spring(Portland) 以前)</a>」に関するコラムを追加</li> <li>▪ 「<a href="#">抽出条件の設定</a>」の説明を修正</li> <li>▪ 「<a href="#">外部データソースの利用</a>」の説明を修正</li> <li>▪ 「<a href="#">REST APIを参照する</a>」を追加</li> </ul>



変更年月日	変更内容
2017-12-01	<p>第16版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> <li>「<a href="#">対応するフィールドの型</a>」にNCLOBを追加</li> <li>「<a href="#">リスト集計におけるCSV出力の設定</a>」にfetch-sizeについての記載を追加</li> <li>「<a href="#">外部データソースの利用</a>」 - 「<a href="#">ロジックフローの利用</a>」を追加</li> </ul>
2018-02-01	<p>第17版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> <li>「<a href="#">設定可能なパラメータ</a>」のdest_storage_pathの説明を修正</li> </ul>
2018-04-01	<p>第18版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> <li>「<a href="#">設定可能なパラメータ</a>」にdest_storage_file_nameの説明を追加</li> <li>「<a href="#">生成されたソースコードを動作させるための手順</a>」にintra-mart Accel Platform 2018 Spring(Skylark)以降用の説明を追加</li> </ul>
2018-08-01	<p>第19版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> <li>「<a href="#">検索の条件値入力に選択型コンポーネントを利用する</a>」を追加</li> </ul>
2019-04-01	<p>第20版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>「<a href="#">リスト集計におけるCSV出力の設定</a>」のinclude-binaryの説明を修正</li> <li>「<a href="#">データ型と送信データについて</a>」を追加</li> <li>「<a href="#">検索設定について</a>」に表示状態の保持についての説明を追加</li> <li>「<a href="#">クロス集計</a>」に「<a href="#">小計行の表示</a>」を追加</li> <li>「<a href="#">クロス集計</a>」に「<a href="#">ソートについて</a>」を追加</li> <li>「<a href="#">クロス集計</a>」に「<a href="#">AND検索とOR検索（各項目を結合する演算子）について</a>」を追加</li> </ul>
2019-08-01	<p>第21版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>「<a href="#">利用可能な関数一覧</a>」のCOLUMNVALUES、LOOKUP、COUNTIF、IF、SUBSTITUTE関数の説明を追加</li> <li>「<a href="#">リスト集計におけるCSV出力の設定</a>」のdate-format、decimal-formatに説明を追加</li> <li>「<a href="#">リスト集計におけるCSV出力の設定</a>」に「<a href="#">フォーマット適用の優先順について</a>」を追加</li> <li>「<a href="#">データ参照の作成</a>」の「<a href="#">リスト集計の作成</a>」に項目説明を追加</li> <li>「<a href="#">データ参照の作成</a>」の「<a href="#">検索設定について</a>」にダウンロード時の説明を追加</li> <li>「<a href="#">JFreeChart</a>」のURLを最新化</li> <li>「<a href="#">Highcharts</a>」のURLを最新化</li> </ul>
2019-12-01	<p>第22版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>「<a href="#">Highcharts</a>」のグラフ画像をHighchartsのバージョンアップに伴い差し替え</li> <li>「<a href="#">Highcharts</a>」の「<a href="#">グラフ設定</a>」にグラフ表示時の注意事項を追加</li> <li>「<a href="#">利用可能な関数一覧</a>」にROWNUMBER関数を追加</li> </ul>

変更年月日	変更内容
2020-04-01	<p>第23版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">Highcharts</a>」のグラフ種類に「<a href="#">エリアチャート</a>」を追加</li> <li>▪ 「<a href="#">Highcharts</a>」のカラム設定にキャプション軸の説明を追加</li> <li>▪ 「<a href="#">Highcharts</a>」のカラム設定でY軸グループの名称を目盛り表示位置に変更、および、説明の修正</li> <li>▪ 「<a href="#">クエリー一覧画面</a>」のクエリー名項目にクエリー名のアイコンの説明を追加</li> <li>▪ 「<a href="#">データ参照の種類を選択</a>」のクエリー一覧にデータ参照作成のアイコンの説明を追加</li> <li>▪ 「<a href="#">作成したデータ参照の閲覧</a>」のデータ参照一覧の説明を画像内から削除し、本文に記載</li> <li>▪ 「<a href="#">ViewCreator について</a>」にSQLビルダの説明を追加</li> <li>▪ 「<a href="#">ViewCreator全体の権限設定</a>」にSQLビルダに関する認可の説明を追加</li> <li>▪ 「<a href="#">クエリーの作成</a>」からクエリー編集画面の詳細の説明を「<a href="#">クエリービルダによるクエリーの作成</a>」と「<a href="#">SQLビルダによるクエリーの作成</a>」に変更</li> <li>▪ 「<a href="#">外部データソースの利用</a>」にSQLビルダにおけるの注意事項を追加</li> <li>▪ 「<a href="#">クエリー編集画面でテーブル単位のアクセス権を設定する</a>」にSQLビルダにおけるの注意事項を追加</li> <li>▪ 「<a href="#">物理Viewの作成と表示</a>」にSQLビルダにおけるの注意事項を追加</li> <li>▪ 「<a href="#">クエリーの作成</a>」に「<a href="#">クエリー一覧に表示されるクエリーについて</a>」を追加</li> </ul>
2020-08-01	<p>第24版 下記を変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">計算式カラムの設定</a>」のヘッダ・フッタの計算式設定画像を差し替え</li> <li>▪ 「<a href="#">リスト集計の作成</a>」のデータ参照編集画面の画像を差し替え</li> <li>▪ 「<a href="#">リスト集計の作成</a>」にタイムゾーン設定の説明を追加</li> <li>▪ 「<a href="#">利用可能な関数一覧</a>」にTIMEZONE関数を追加</li> </ul>
2020-12-01	<p>第25版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">ViewCreator全体の権限設定</a>」に認可リソース「<a href="#">ルーティングの作成</a>」についての記載を追加</li> <li>▪ 「<a href="#">インポート・エクスポート</a>」のインポート・エクスポートに関する注意事項を修正</li> <li>▪ 「<a href="#">REST APIを参照する</a>」のREST APIの説明および画像を追加</li> <li>▪ 「<a href="#">ルーティングの作成</a>」を追加</li> </ul>
2021-04-01	<p>第26版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">ViewCreator全体の権限設定</a>」の認可リソース「<a href="#">インポート・エクスポート</a>」に管理対象機能を追加</li> <li>▪ 「<a href="#">インポート・エクスポート</a>」に追加機能の説明を追加</li> <li>▪ 「<a href="#">ルーティングの作成</a>」の「<a href="#">ルーティングの実行確認</a>」の説明を修正</li> <li>▪ 「<a href="#">ルーティングの作成</a>」に「<a href="#">REST APIの仕様確認</a>」の説明を追加</li> <li>▪ 「<a href="#">外部データソースの利用</a>」の「<a href="#">ロジックフローを検索する</a>」にフローカテゴリと検索結果の表示の説明を追加</li> </ul>
2021-08-01	<p>第27版 下記を追加・変更しました</p> <ul style="list-style-type: none"> <li>▪ 「<a href="#">検索設定について</a>」に日付型、タイムスタンプ型の検索設定に関する説明を追加</li> </ul>

ViewCreator はintra-martのWeb画面上から、データベースのデータを使用して、表やグラフを簡単に作成できるツールです。  
テナントデータベースとシェアドデータベース、どちらも使用可能です。

ViewCreator では、大別して2種類のデータを扱います。

### ■ クエリ

データベース上のテーブルを使用して作成されるSQLクエリです。

ドラッグ&ドロップなどを用いたGUI操作による作成と、SQLの直接入力による作成が行えます。

### コラム

SQLの直接入力による作成は2020 Spring(Yorkshire)から利用可能です。

### ■ データ参照

クエリの表示方法（表orグラフなど）の設定です。

1つのクエリから複数の見せ方（データ参照）をさせることが可能です。

またデータ参照には閲覧権限を設定することもできます。

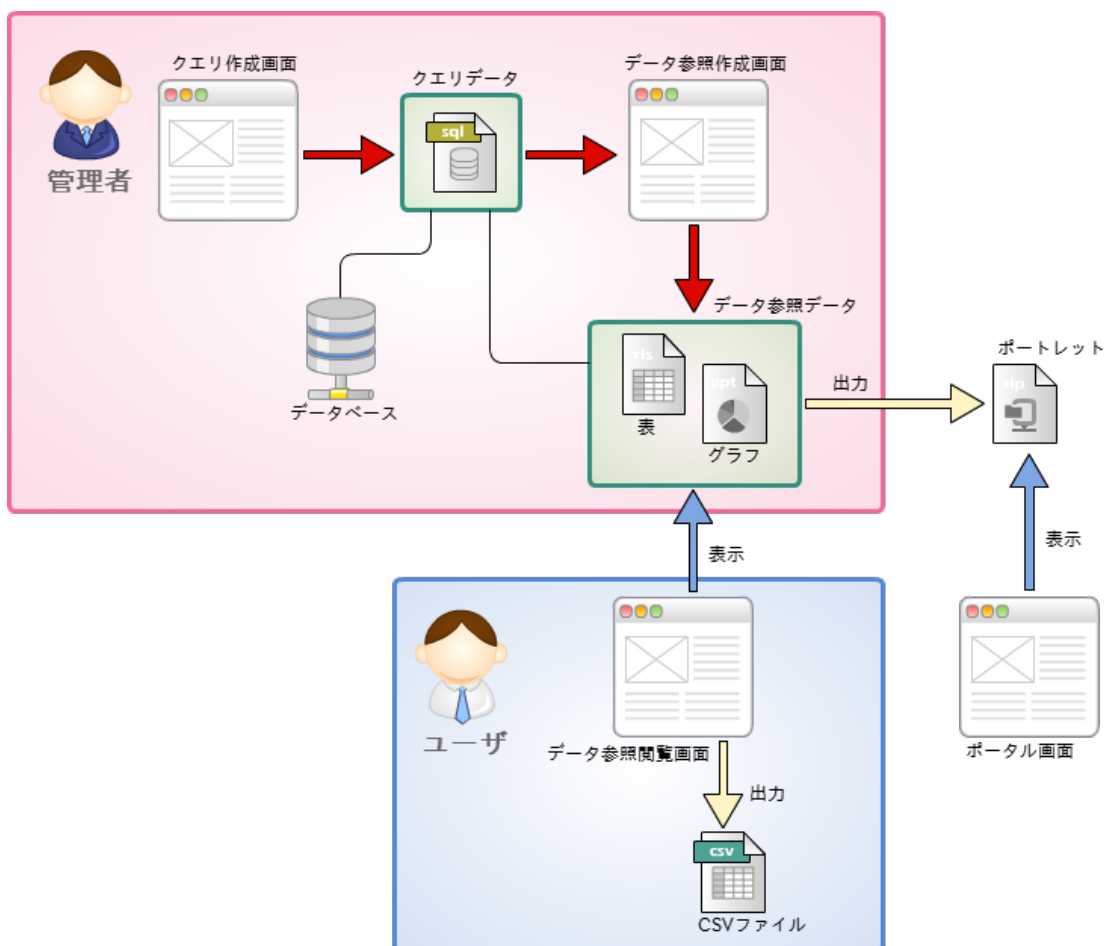


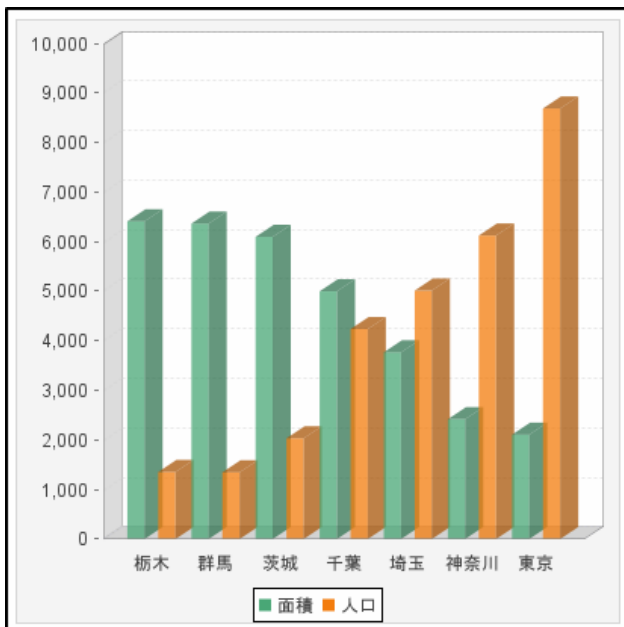
図 : ViewCreator 概要図

管理者はまず、既存のテーブル情報を用いてクエリを作成します。

クエリを作成したら、それを元にデータ参照を作ります。

年度	地域名	都道府県名	面積	人口
1985	近畿	+ 京都	-	-
		+ 三重	-	-
		+ 滋賀	-	-
		+ 大阪	-	-
		+ 奈良	-	-
		+ 兵庫	-	-
		+ 和歌山	-	-

図：データ参照の例1 - グループ化された表



図：データ参照の例2 - 棒グラフ

作成したデータ参照は、そのままユーザに見せるだけでなく  
 ポートレットに出力して、ポータル画面に表示することもできます。  
 また、リスト形式のデータ参照は表示しているデータをCSVファイルとして出力できます。

## 対応するフィールドの型

ViewCreator機能で対応する各種データベースごとの型は次の通りです。

データベース名	文字列型	数値型	日付型	タイムスタンプ		
				型	バイナリ型	真偽値型
Oracle	VARCHAR2 VARCHAR NVARCHAR2 CHAR NCHAR	NUMBER BINARY_FLOAT BINARY_DOUBLE	DATE (*mapDateToTimestamp をfalseに設定している場合のみ)	DATE TIMESTAMP	BLOB CLOB NCLOB	なし
SQL Server	varchar char nvarchar nchar ntext	bigint int smallint tinyint decimal numeric money smallmoney float real	date	datetime smalldatetime datetime2	binary varbinary	bit

データベース名	文字列型	数値型	日付型	タイムスタンプ型	バイナリ型	真偽値型
PostgreSQL	varchar character text	smallint bigint decimal numeric real	date	timestamp	bytea	boolean
SAP HANA	VARCHAR NVARCHAR CHAR NCHAR	BIGINT SMALLINT INTEGER DOUBLE DECIMAL TINYINT REAL	DATE	TIMESTAMP	BLOB CLOB NCLOB	BOOLEAN

 コラム

真偽値型は2014 Summer(Honoka)から利用可能です。

 注意

バイナリ型のフィールドはリスト集計のみ対応します。  
また、データ参照時にバイナリ型のフィールドは空欄で表示されます。  
CSV出力時のみ、実データの出力が行われます。

 注意

ViewCreator上でのデータ型は、JDBCドライバの実装やデータベース製品の仕様に依存します。  
そのため、ドライバのバージョンによって上記の表の通りのマッピングにならない可能性があります。

ここでは基本的な操作について紹介します。

## ViewCreator全体の権限設定

まず最初に一般ユーザがViewCreatorのメニューを使えるように設定します。

### 目次

- 認可の設定
- メニューの権限設定

### 認可の設定

1. 「サイトマップ」をクリックします。
2. 「テナント管理」→「認可」をクリックします。

リソース	アクション	認証		組織		ロール								
		ゲストユーザ	認証済みユーザ	サンプル会社	その他会社	テナント管理者	認可管理者	メニュー管理者	メニュー運用管理者	アカウント管理者	ロール管理者	カレンダー管理者	ジョブスケジューリング管理者	
画面・処理														
テーブル・キャプション登録	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
ViewCreator	実行	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
データ参照の閲覧・作成	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
外部データソース連携	実行	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
ロジックフロー管理	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
ファイルリソースの追加	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
インポート・エクスポート	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
管理者	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
クエリの作成	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Viewの作成	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
WHERE句の直接入力	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
ルーティングの作成	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
SQLビルダ	実行	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗

リソース名	説明
データ参照の閲覧・作成	<p>データ参照一覧へのアクセスを可能にします。</p> <p>ただし、閲覧権はデータ参照を作成したユーザが個別に設定可能です。</p> <p>データ参照の作成はログインユーザが作成したクエリが少なくとも1つ以上登録されていなければ実行できません。</p> <p>つまり、データ参照の作成はクエリの作成を許可されたユーザのみが利用可能です。</p> <p>以上の理由から一般的には、このリソースに関しては比較的緩い権限設定で問題ありませんが、個々のケースに従って適切なアクセス権を設定してください。</p>
ファイルリソースの追加	<p>外部ファイルリソースの追加・削除を可能にします。</p> <p>一般的には管理者ユーザのみを使用可能に設定します。</p>

リソース名	説明
インポート・エクスポート	ViewCreatorの設定データを一括でインポート/エクスポートする機能へのアクセスを可能にします。 一括でインポート/エクスポートする機能として以下の機能が管理されます。 ・ クエリとデータ参照のインポート/エクスポート ・ インポート ・ エクスポート 個別でインポート/エクスポートする機能は管理外です。  一般的には管理者ユーザのみを使用可能に設定します。
管理者	ViewCreatorを管理者として利用できるようにします。 具体的には、管理者ユーザは登録されている全てのクエリやデータ参照に対して変更や削除が可能です。 ただし、クエリ一覧画面やデータ参照一覧画面を利用するためには「クエリの作成」、 「データ参照の閲覧・作成」への認可設定が必要です。
クエリの作成	このリソースへのアクセスを許可されたユーザはクエリの作成とデータ参照の作成が可能です。  標準設定では全テーブルの全データが閲覧可能になるため、一般的に管理者ユーザ以外の利用を許可する場合は参照可能なテーブルを絞る必要があります。 詳細は「 <a href="#">クエリ編集画面でテーブル単位のアクセス権を設定する</a> 」を参照してください。
Viewの作成	クエリ編集画面において、「Viewの作成」機能の利用が可能です。 この機能を利用すると、任意のSQLの発行が可能です。 以上の理由から、かなり制限されたユーザだけに許可するようにしてください。
WHERE句の直接入力	クエリ編集画面において、抽出条件として任意のWHERE句を入力できます。 Viewの作成と同様に、かなり制限されたユーザだけに許可するようにしてください。
ルーティングの作成	このリソースへのアクセスを許可されたユーザはルーティングの作成が可能です。 この機能を利用すると、クエリを直接実行するルーティングURLの作成が可能です。 以上の理由から、かなり制限されたユーザだけに許可するようにしてください。 詳細は「 <a href="#">ルーティングの作成</a> 」を参照してください。
SQLビルダ	クエリ編集画面において、任意のSELECT文を直接入力してクエリの作成/編集が行える「SQLビルダ」画面の利用が可能です。 ただし、「SQLビルダ」画面の利用を行うためには「クエリの作成」の認可設定も合わせて必要です。 「SQLビルダ」画面は、テーブル単位の認可の制御は行われません。 そのため、かなり制限されたユーザだけに許可するようにしてください。 テーブル単位の認可の制御についての詳細は「 <a href="#">クエリ編集画面でテーブル単位のアクセス権を設定する</a> 」を参照してください。

### コラム

Viewの作成、WHERE句の直接入力は2014 Winter(Iceberg)から利用可能です。

管理者は2015 Summer(Karen)から利用可能です。

SQLビルダは2020 Spring(Yorkshire)から利用可能です。

ルーティングの作成は2020 Winter(Azalea)から利用可能です。

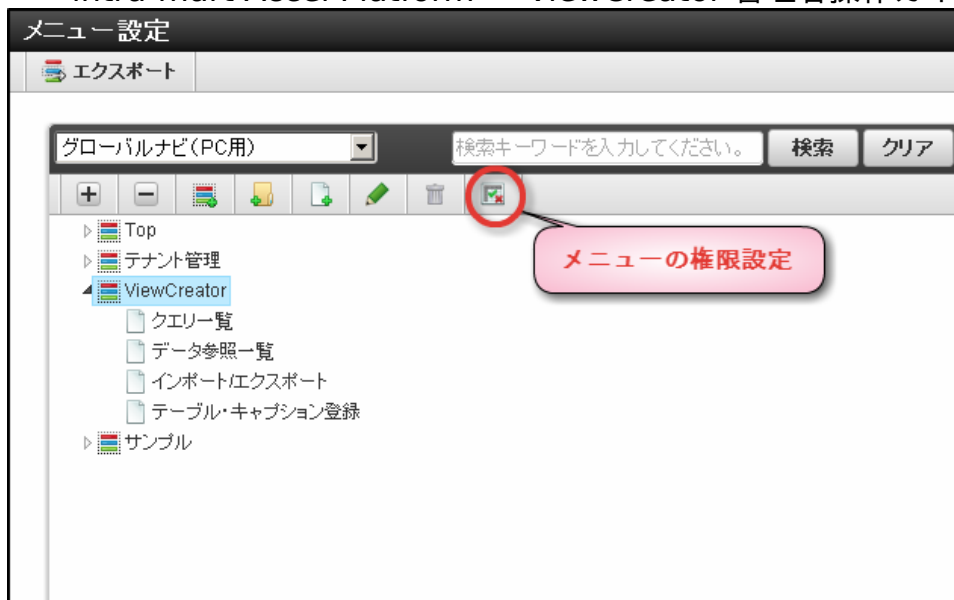
インポート/エクスポートは2021 Spring(Bergamot)からクエリとデータ参照のインポート/エクスポートに名称が変更されました。

インポートは2021 Spring(Bergamot)から利用可能です。

エクスポートは2021 Spring(Bergamot)から利用可能です。

## メニューの権限設定

1. 「サイトマップ」をクリックします。
2. 「テナント管理」→「メニュー」をクリックします。



認可と同様に設定します。設定したユーザのメニューに「ViewCreator」の項目が表示され、使用できます。

## クエリの作成

接続するデータベースを指定し、クエリを新規に作成します。

クエリは、ドラッグ&ドロップなどを用いたGUI操作による作成と、SQLの直接入力による作成が行えます。

### i コラム

HTML版のクエリ編集画面は 2017 Summer(Quadra) から利用可能です。

2017 Spring(Portland) 以前のバージョンのクエリ作成については「[クエリの作成 \(2017 Spring\(Portland\) 以前\)](#)」も合わせて参照してください。

また、2017 Summer(Quadra) 以降のバージョンでFlash版のクエリ編集画面を利用することも可能です。

詳細は「[設定ファイルリファレンス](#)」-「[クエリ編集画面設定](#)」を参照してください。

### i コラム

SQLの直接入力による作成は2020 Spring(Yorkshire)から利用可能です。

Flash版のクエリ編集画面ではSQLの直接入力によるクエリの作成は行えません。

### 目次

- [クエリー一覧画面](#)
  - [クエリー一覧に表示されるクエリについて](#)
- [クエリ編集画面](#)

## クエリー一覧画面

「サイトマップ」→「ViewCreator」→「クエリー一覧」をクリックします。





図：クエリー一覧画面

項目名	説明
新規	クエリ定義の新規作成を行う「クエリ編集」画面に遷移します。
コピー	クエリ定義をコピーします。 コピーしたいクエリのチェックボックスにチェックをいれ、「コピー」をクリックしてください。
削除	クエリ定義を削除します。 削除したいクエリのチェックボックスにチェックをいれ、「削除」をクリックしてください。
データ参照一覧へ	「データ参照一覧」画面に遷移します。
検索	クエリ定義を検索します。 テキストボックスに設定されたキーワードとクエリ名の部分一致で検索します。
データ参照作成	各アイコンからデータ参照の作成画面に遷移します。 詳細は「 <a href="#">データ参照の作成</a> 」を参照してください。
クエリー名	クリックすることで該当のクエリ定義の「クエリ編集画面」に遷移します。 クエリ定義を作成する方法によって、以下のいずれかのアイコンが表示されます。  <b>クエリビルダ</b> で作成したクエリです。  <b>SQLビルダ</b> で作成したクエリです。
エクスポート	クエリ定義のエクスポートを行います。 エクスポートするクエリ定義の「エクスポート」アイコンをクリックしてください。
インポート	クエリ定義のインポートを行います。 インポートするクエリを「ファイル追加」で選択し、「インポート」をクリックしてください。

### コラム

作成したクエリはXML形式のファイルでインポート/エクスポートができます。

### 注意

エクスポートされたXMLデータにはデータベースの仕様に依存する可能性があるデータ（テーブル名とフィールド名）が含まれます。  
大文字、小文字の違い等によりインポートしたクエリが正しく表示できない場合がありますので注意してください。

## クエリー一覧に表示されるクエリについて

クエリー一覧に表示されるクエリは、それぞれのクエリごとに参照可能かを判定し、参照可のクエリのみ表示しています。判定条件は以下の通りです。

クエリ内のテーブルのアクセス権については「[クエリ編集画面でテーブル単位のアクセス権を設定する](#)」を参照してください。

2019 Winter(Xanadu)まで

対象者	参照可能なクエリ
管理者	全てのクエリ
管理者以外	該当ユーザが作成し、かつ、更新者もそのユーザであるクエリ または クエリ内の全テーブルに参照許可が付与されているクエリ

2020 Spring(Yorkshire)から

対象者	参照可能なクエリ
管理者	全てのクエリ
管理者以外	クエリ内の全テーブルに参照許可が付与されているクエリ

## クエリ編集画面

クエリー一覧画面の「新規」をクリックして作成画面を開きます。

クエリ編集画面は以下の2種類の画面があります。

- **クエリビルダ**

ドラッグ&ドロップなどを用いたGUI操作によるクエリの作成が行えます。

「新規」をクリックした場合、初期表示としてクエリビルダが表示されます。

- **SQLビルダ**

SQLの直接入力によるクエリの作成が行えます。

クエリビルダのツールバーに表示される「SQLで編集」をクリックしてSQLビルダに切り替えることが可能です。



#### コラム

SQLビルダは2020 Spring(Yorkshire)から利用可能です。

## クエリビルダによるクエリの作成

クエリビルダではドラッグ&ドロップなどを用いたGUI操作でクエリの作成が行えます。

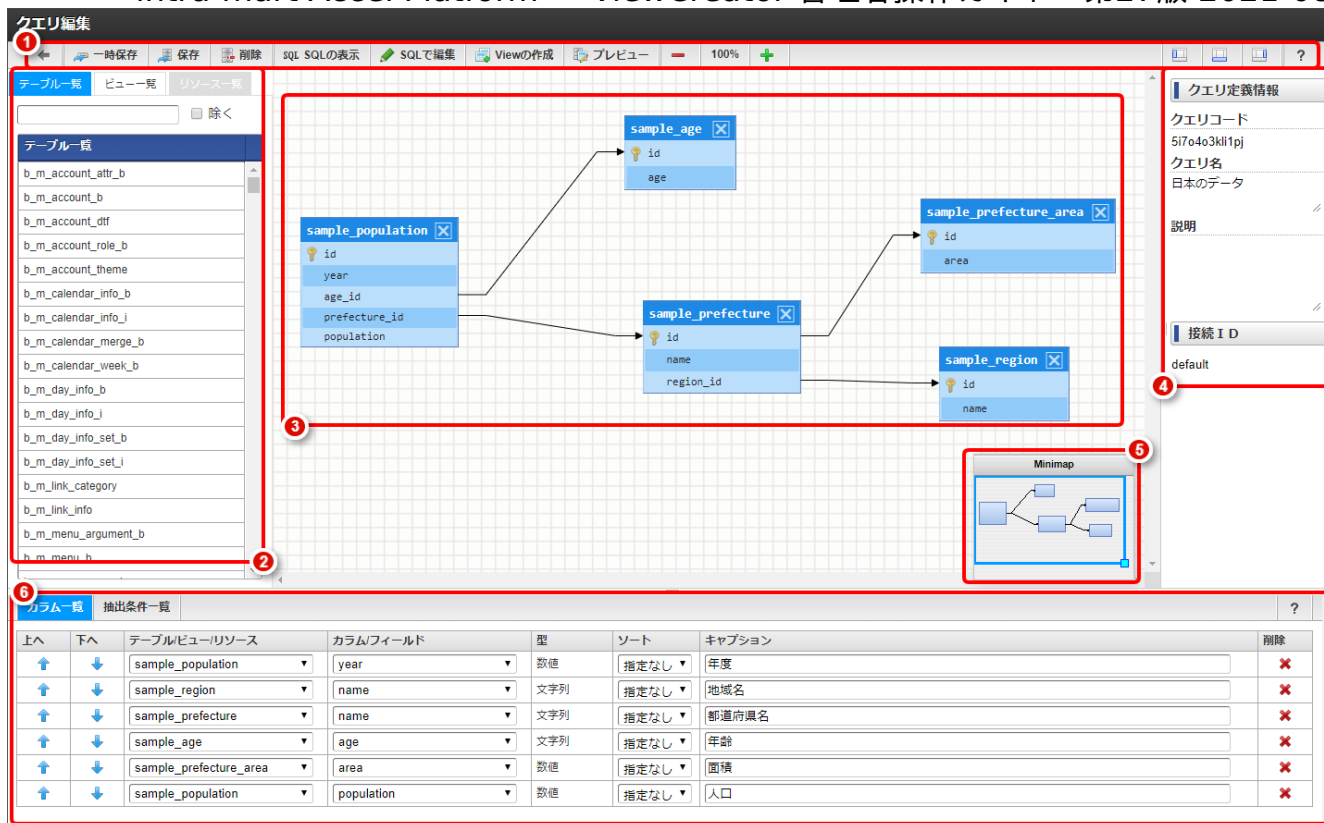
#### 目次

- [クエリ編集画面](#)
- [テーブルの追加](#)
- [結合条件の作成](#)
- [カラムの追加](#)
- [SQLの表示とプレビュー](#)
- [クエリを保存する](#)

## クエリ編集画面

クエリー一覧画面の「新規」をクリックして作成画面を開きます。

クエリ編集画面の詳細は以下の通りです。



図：クエリ編集画面 - 初期表示

- クエリ編集に対する基本的な操作を提供するヘッダです。  
表示項目は以下の通りです。

項目名	説明
一時保存	クエリ定義をクライアントローカルに一時保存します。 保存可能なクエリ定義の個数は1つだけです。 クエリ編集画面を開くときに一時保存されているクエリ定義を読み込むことが可能です。
保存	クエリ定義を登録します。
削除	クエリ定義を削除します。 新規作成時には「削除」ボタンは表示されません。
SQLの表示	クエリ定義からSQL文を生成してダイアログで表示します。
SQLで編集	クエリの作成方法をSQLビルダへと変更します。 変更時には現在のクエリ定義から生成したSQL文がSQLビルダのエディタ上に表示されます。 なお、抽出条件として動的パラメータを利用している場合は展開されて表示されます。
Viewの作成	クエリ定義から生成されるSQL文をベースにしたCREATE VIEW文を生成してダイアログで表示します。 また、作成したViewはテーブルと同じようにクエリに組み込むことが可能です。 詳細は「 <a href="#">物理Viewの作成と表示</a> 」を参照してください。
プレビュー	クエリ定義から生成されるSQL文の実行結果をダイアログで表示します。
デザイナの拡大・縮小	デザイナの表示倍率を変更します。 100%をクリックすると初期表示時の倍率にリセットされます。
エンティティ一覧	エンティティ（テーブル・ビュー・リソース）一覧の表示/非表示を切り替えます。
カラム/抽出条件一覧	カラム一覧、抽出条件一覧の表示/非表示を切り替えます。
プロパティ	プロパティ（クエリ定義情報・接続ID）の表示/非表示を切り替えます。
?	サイトツアーが呼び出されます。

**i** コラム

一時保存後に保存を行わずにクエリ編集画面を閉じた場合は、該当のクエリ定義の編集を行う際に以下のダイアログが表示されます。



ダイアログは保存または更新されるまで、編集画面を開く度に表示されます。

**i** コラム

コラム一覧および抽出条件一覧のサイトツアーは選択中のタブのみ表示されます。

もう一方のタブのサイトツアーを呼び出す際は、タブを変更し「コラム/抽出条件一覧」のヘッダに表示されている「？」ボタンをクリックしてください。

コラム一覧または抽出条件一覧のサイトツアーのみ呼び出されます。

**i** コラム

SQLビルダは2020 Spring(Yorkshire)から利用可能です。

また、「SQLで編集」のリンクは「SQLビルダ」か「管理者」の認可が無ければ表示されません。

認可についての詳細は「[ViewCreator全体の権限設定](#)」を参照してください。

**!** 注意

クエリビルダで作成し保存したクエリを、SQLビルダに切り替えた場合、クエリビルダに戻すことはできません。

クエリビルダで作成し保存したクエリを残したい場合は、SQLビルダで新規クエリとして保存を行うか、コピー機能を利用してください。

**!** 注意

すでにデータ参照で利用しているクエリの場合は、SQLビルダでの編集に切り替えることはできません。

2. 接続IDに紐づくエンティティの一覧を表示します。  
タブを切り替えることでテーブル一覧、ビュー一覧、リソース一覧が表示されます。  
ビューの詳細は「[物理Viewの作成と表示](#)」を参照してください。  
リソースの詳細は「[外部データソースの利用](#)」を参照してください。
3. テーブルやビューを配置し結合条件などを設定するデザイナー部分です。
4. クエリの名称や説明、接続IDなどのプロパティを表示します。  
クエリ定義情報は保存ダイアログで入力を行います。

**i** コラム

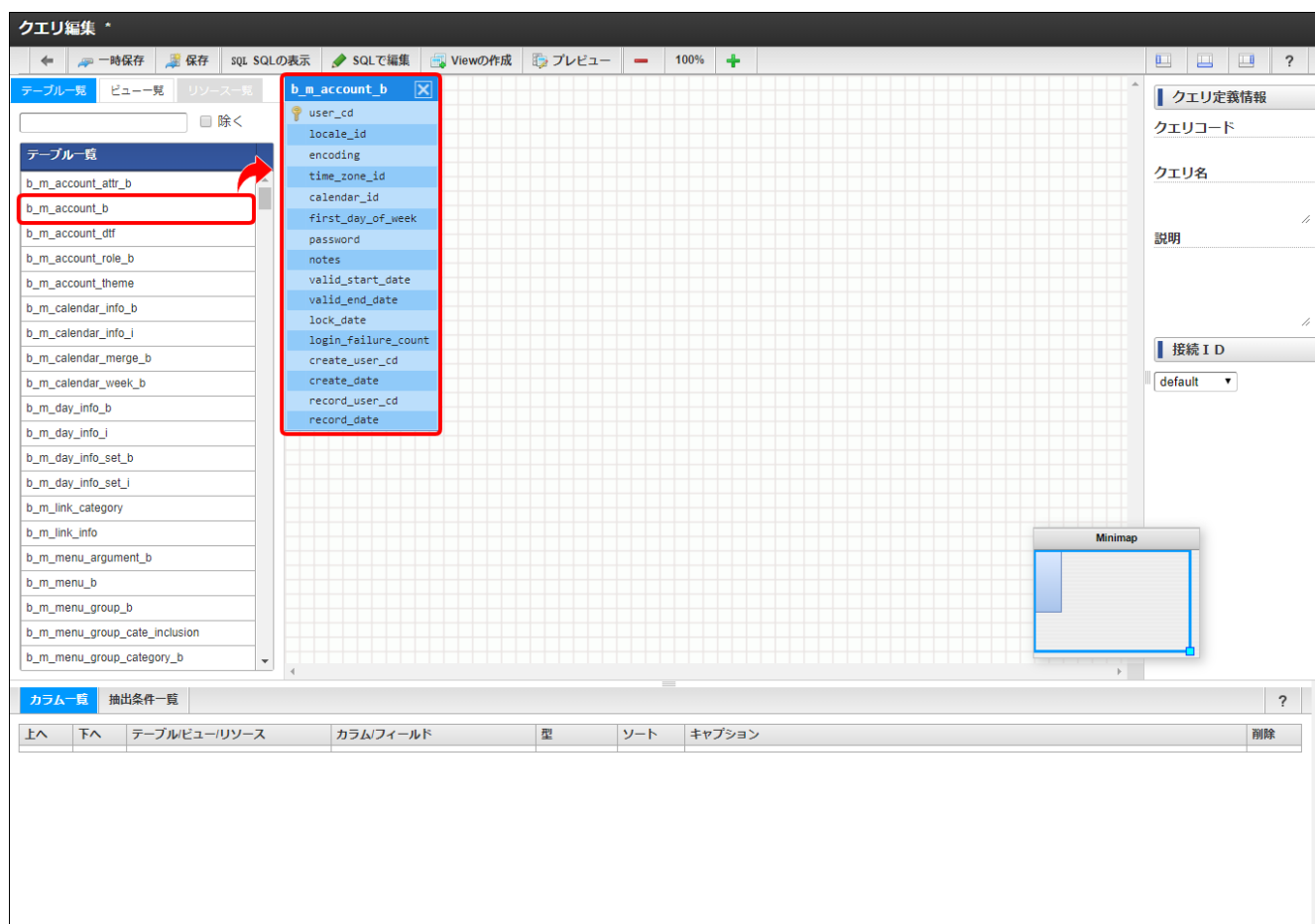
SQLビルダへの切り替えはテナントデータベースかシェアデータベースを利用している場合のみ可能です。

外部データソースを利用している場合はSQLビルダへの切り替えはできません。

5. デザイナー部分の全体図を表します。  
青い枠に囲まれた部分が現在の描写範囲で、ドラッグすることで描写範囲を変更できます。
6. クエリのコラムや抽出条件を設定する部分です。  
タブを切り替えることでコラム一覧および抽出条件一覧を表示します。

## テーブルの追加

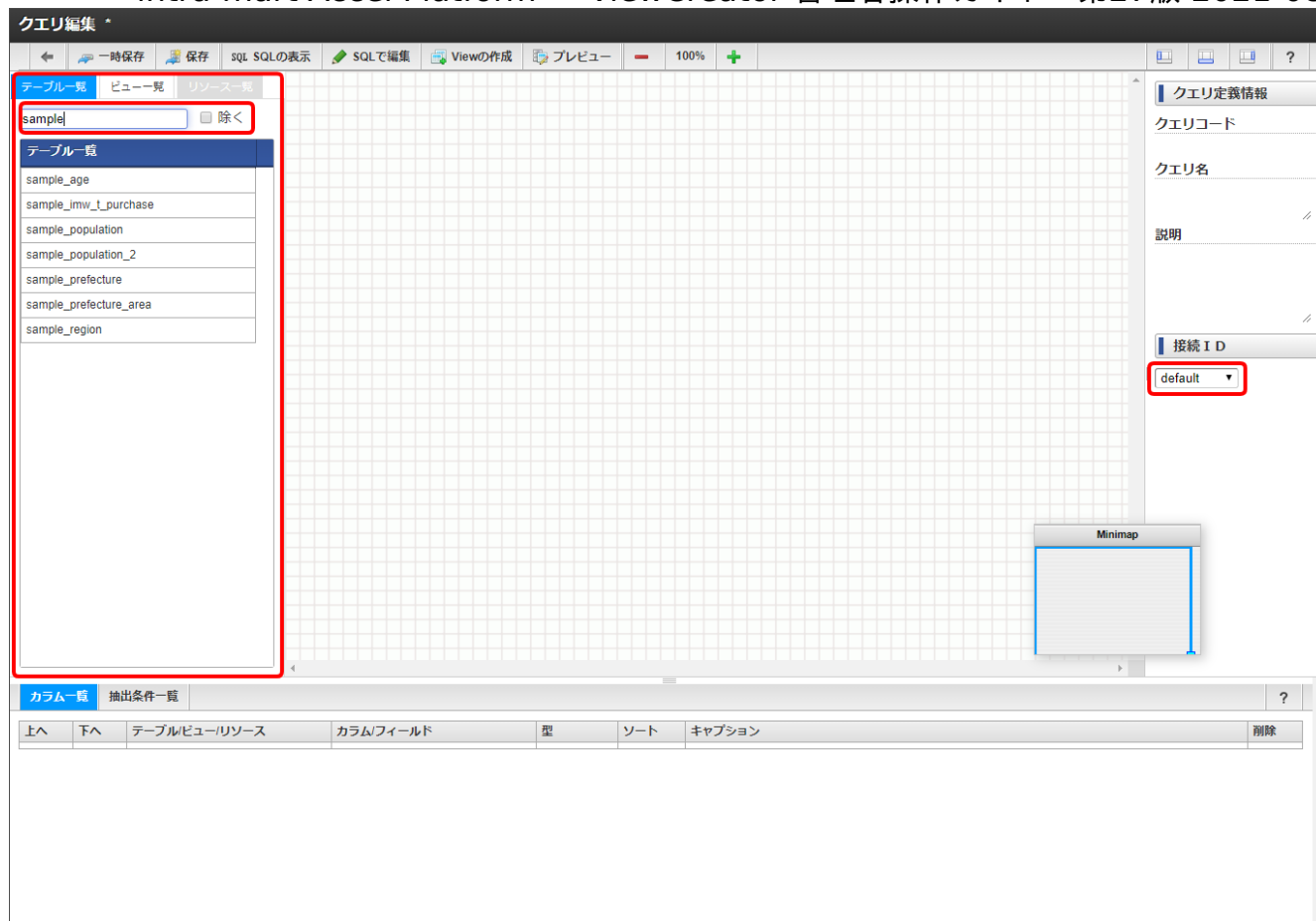
「テーブル一覧」から追加するテーブルをダブルクリックすることで、デザイナーにテーブルウィンドウが表示されます。  
「カギ」アイコンは主キーを表します。



図：クエリ編集画面 - テーブル追加

テーブルおよびビューの一覧は、接続IDに紐づくテーブルの一覧を表示します。  
ただし、intra-mart Accel Platform が内部のみで利用する一部のテーブルは一覧に表示されません。

また、検索キー入力により表示するテーブルとビューを絞り込むことができます。  
「除く」を選択した場合は、記入した文字を含まないテーブルとビューが一覧に表示されます。



図：クエリ編集画面 - テーブル一覧

### **i** コラム

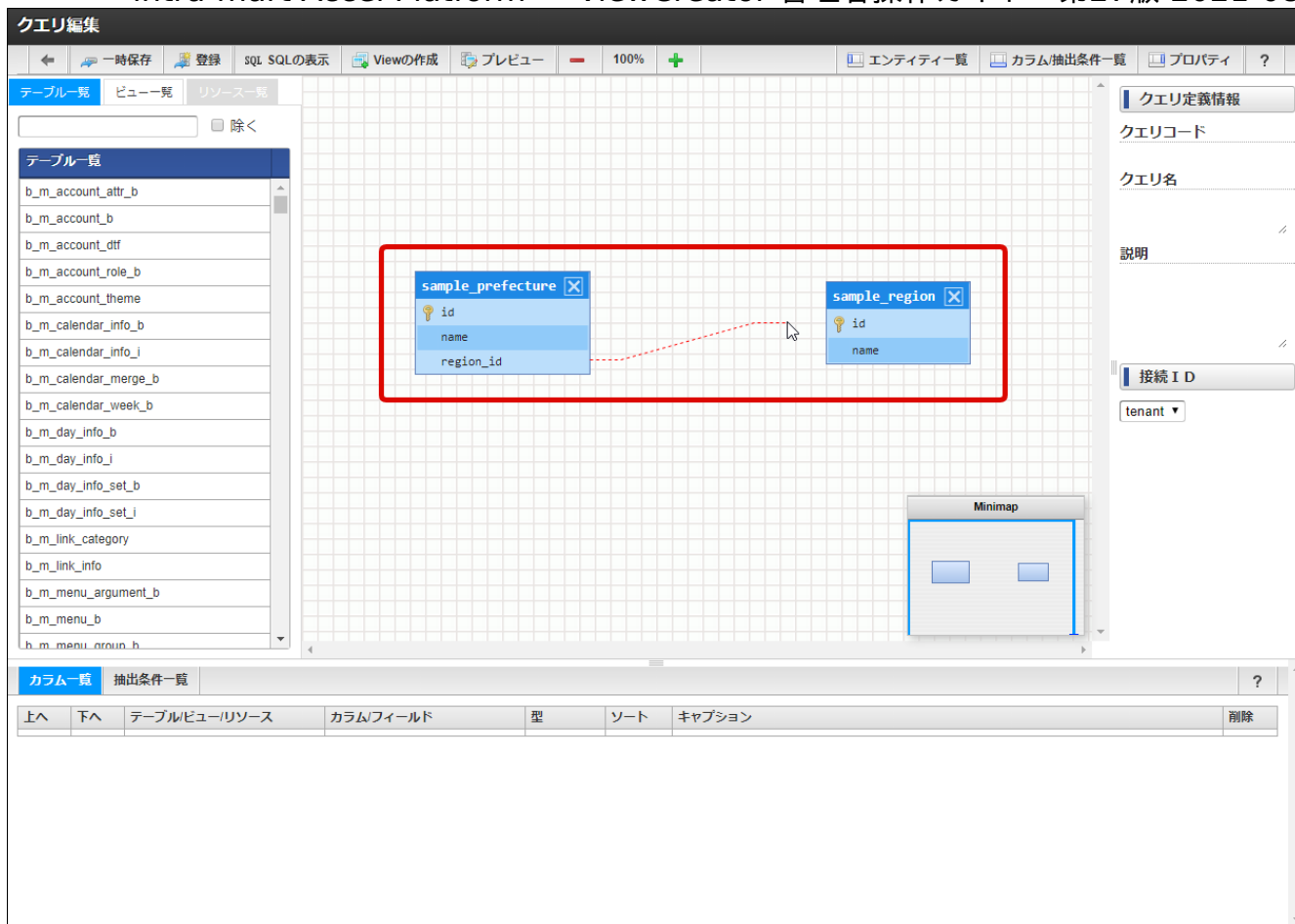
クエリのエクスポートデータ(XML)内の<db-source>タグの内容を書き換えることで接続IDの変更に対応できます。

### **!** 注意

複数の接続IDをまたがったクエリは作成できません。

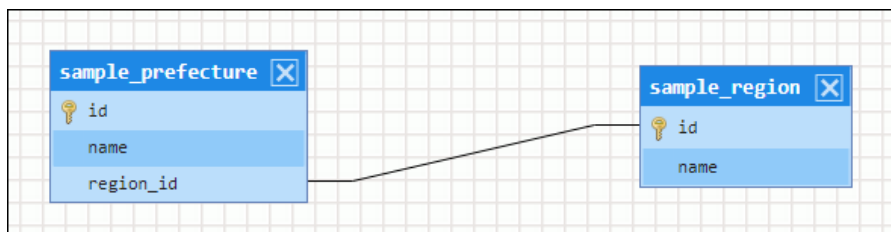
## 結合条件の作成

任意のフィールドから別のエンティティのフィールドへドラッグ&ドロップすることで結合条件を作成できます。



図：クエリ編集画面 - 結合条件

結合条件（結合線）が描画されます。

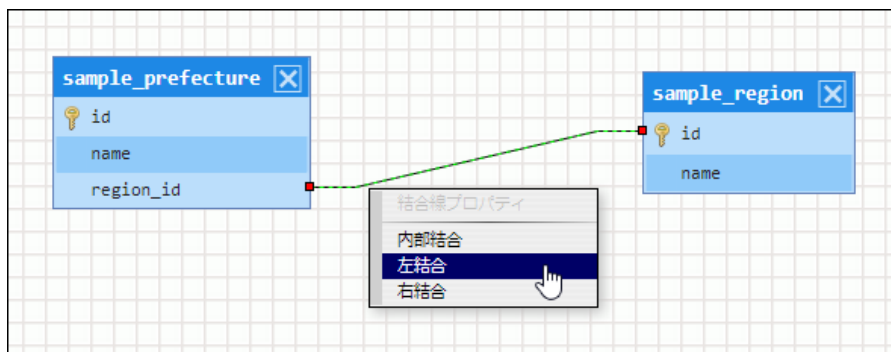


図：クエリ編集画面 - 結合線

結合線を右クリックすると左結合(LEFT OUTER JOIN)や右結合(RIGHT OUTER JOIN)に変更できます。

※初期設定では内部結合(INNER JOIN)です。

また、選択中の結合線はDELETEキーで削除できます。



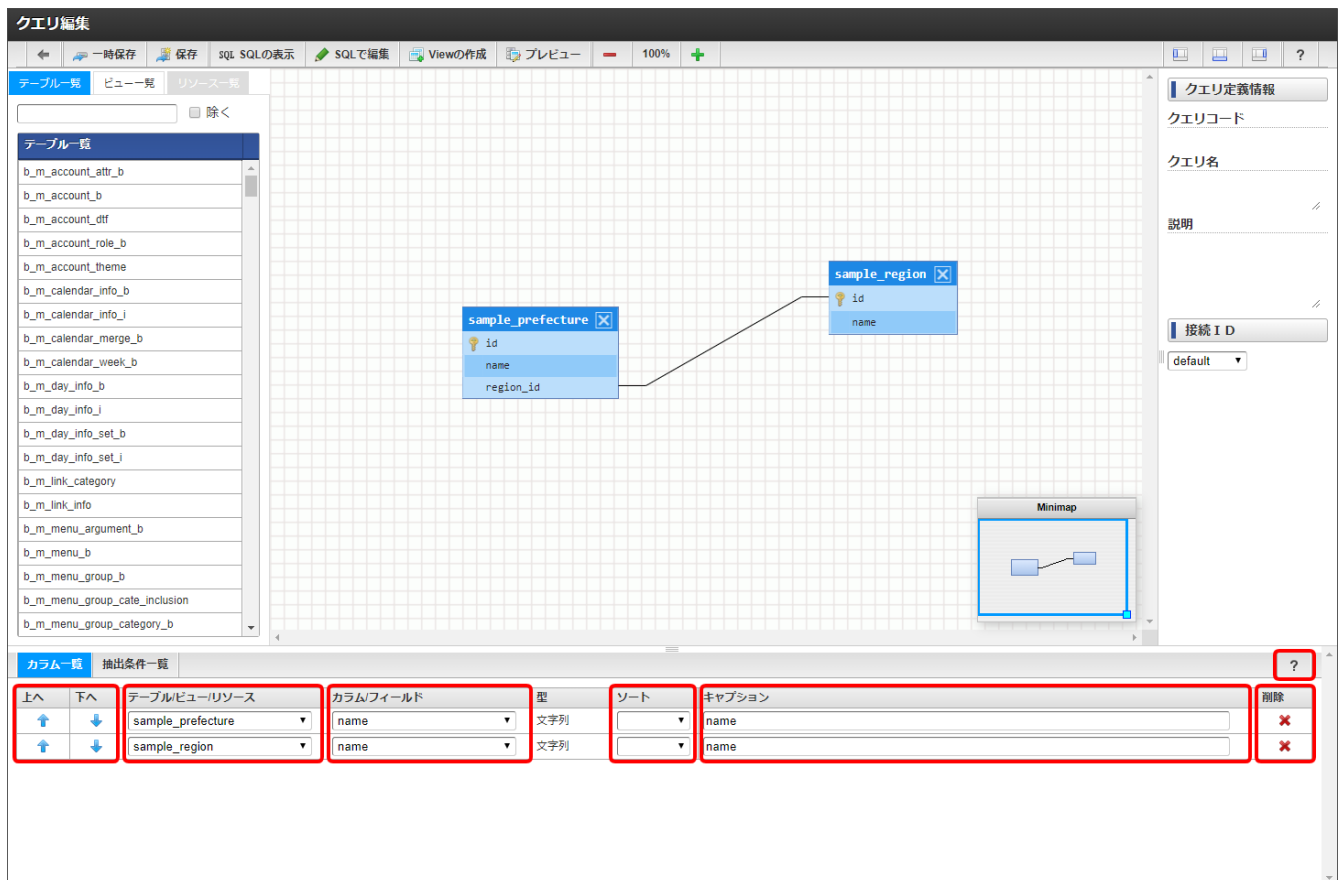
図：クエリ編集画面 - 結合条件変更

**注意**

エンティティがリソースの場合は、結合条件を作成できません。

## カラムの追加

フィールドをダブルクリックすると、そのフィールドをクエリのカラムとして追加できます。



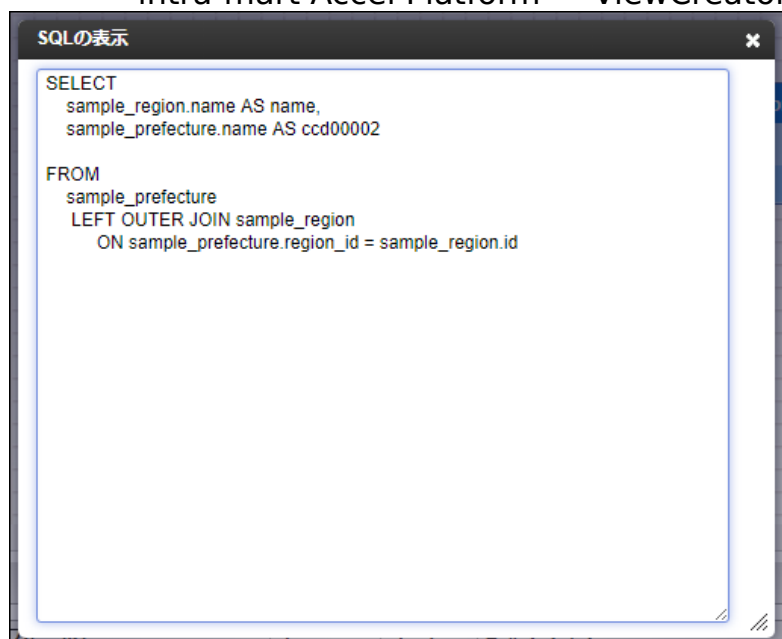
図：クエリ編集画面 - カラム追加

項目名	説明
上へ/下へ	カラムの表示順を入れ替えます。
テーブル/ビュー/リソース	デザイナーに追加されているエンティティを選択可能です。
カラム/フィールド	クエリで利用するカラムを選択します。 「テーブル/ビュー/リソース」で選択されたエンティティが持つカラムを選択可能です。
ソート	クエリからグラフ集計を作成した場合に適用されます。 それ以外の場合は、データ参照で設定されたソート順が適用されます。
キャプション	カラムに対するラベル名を設定します。 データ参照作成時に項目名の初期値としてセットされる他、計算式において変数名として利用できます。
削除	カラム一覧からカラムを削除します。
?	選択中のタブ（カラム一覧または抽出条件一覧）のサイトツアーが呼び出されます。

## SQLの表示とプレビュー

画面上部の「SQLの表示」を選択すると作成したクエリのSQL文、「プレビュー」を選択すると作成したクエリの実行結果を確認できます。





図：クエリ編集画面 - SQLの表示

地域名	都道府県名
北海道	北海道
東北	青森
東北	岩手
東北	宮城
東北	秋田
東北	山形
東北	福島
関東	茨城
関東	栃木
関東	群馬
関東	埼玉
関東	千葉
関東	東京
関東	神奈川
中部	新潟
中部	富山

図：クエリ編集画面 - プレビュー

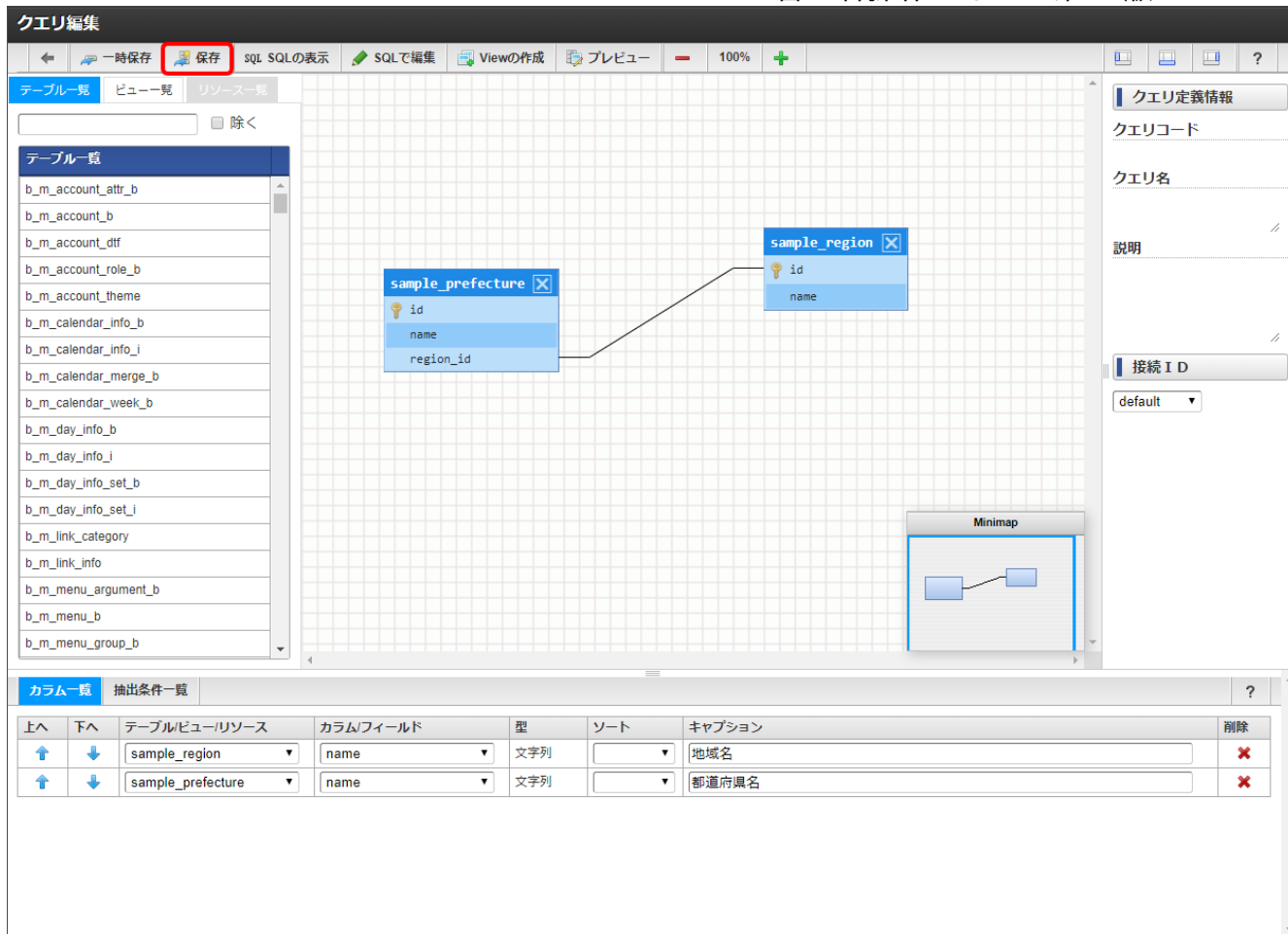
### コラム

タイムスタンプ型のコラムは、プレビュー時はISO 8601形式の日時フォーマットで表示されます。

例：2017-08-01T12:34:56.789Z

## クエリを保存する

画面上部の「保存」ボタンで作成したクエリを登録します。



図：クエリ編集画面 - 保存

クエリ定義情報の保存または更新ダイアログが表示されます。



図：クエリ編集画面 - 保存ダイアログ

項目名	説明
クエリコード	クエリ定義を識別する一意のコードを設定します。 クエリコードは必須項目で、利用可能な文字は半角英数字とアンダースコアです。 新規作成時のみ入力可能で、編集はできません。
クエリ名	クエリ名を設定します。 必須項目です。
説明	クエリ定義の説明を設定できます。 任意項目です。

**注意**

既存のクエリの編集は、管理者ユーザかそのクエリを作成したユーザのみが実行できます。

SQLビルダではSQLを直接入力してクエリの作成が行えます。

#### 目次

- クエリ編集画面
- カラムの追加
- プレビュー
- クエリを保存する

## クエリ編集画面

クエリビルダの「SQLで編集」をクリックしてSQLビルダの作成画面を開きます。  
クエリ編集画面の詳細は以下の通りです。

The screenshot shows the 'Query Editor' (クエリ編集) interface. It features a main SQL editor area with a red box highlighting the SQL code (1), a right-hand panel for 'Query Definition Information' (クエリ定義情報) with fields for code, name, and description (3), and a bottom section for 'Column List' (カラム一覧) with a 'Refresh' button (4). The column list table is as follows:

カラム/フィールド	型	キャプション
year	数値	年度
region_name	文字列	地域名
prefecture_name	文字列	都道府県名
age	文字列	年齢
area	数値	面積

図：クエリ編集画面 - 初期表示

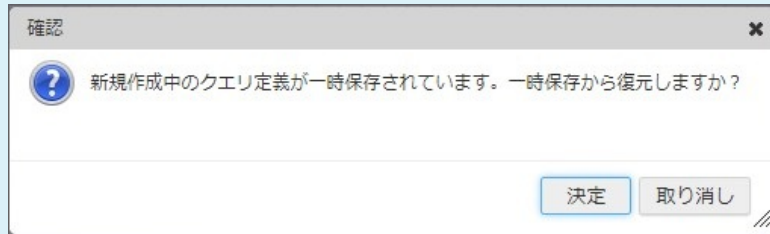
- クエリ編集に対する基本的な操作を提供するヘッダです。  
表示項目は以下の通りです。

項目名	説明
一時保存	クエリ定義をクライアントローカルに一時保存します。 保存可能なクエリ定義の個数は1つだけです。 クエリ編集画面を開くときに一時保存されているクエリ定義を読み込むことが可能です。
保存	クエリ定義を登録します。 正常なSQLを記述し、カラム一覧に反映をクリックした後でないでクエリの登録はできません。
削除	クエリ定義を削除します。 新規作成時には「削除」ボタンは表示されません。

項目名	説明
プレビュー	作成したクエリの実行結果をダイアログで表示します。 正常なSQLを記述し、カラム一覧に反映をクリックした後でない限りプレビューの表示はできません。
カラム/抽出条件一覧	カラム一覧、抽出条件一覧の表示/非表示を切り替えます。
プロパティ	プロパティ（クエリ定義情報・接続ID）の表示/非表示を切り替えます。
?	サイトツアーが呼び出されます。

### コラム

一時保存後に登録または更新を行わずにクエリ編集画面を閉じた場合は、該当のクエリ定義の編集を行う際に以下のダイアログが表示されます。



ダイアログは登録または更新されるまで、編集画面を開く度に表示されます。

- SQLを記述するエディタ部分です。

### 注意

SQLはSELECT文のみ利用可能です。  
また、SQLはサブクエリとして実行されます。ご利用のデータベースに合わせてSQLを記述してください。

- クエリの実名や説明、接続IDなどのプロパティを表示します。  
クエリ定義情報は保存ダイアログで入力を行います。

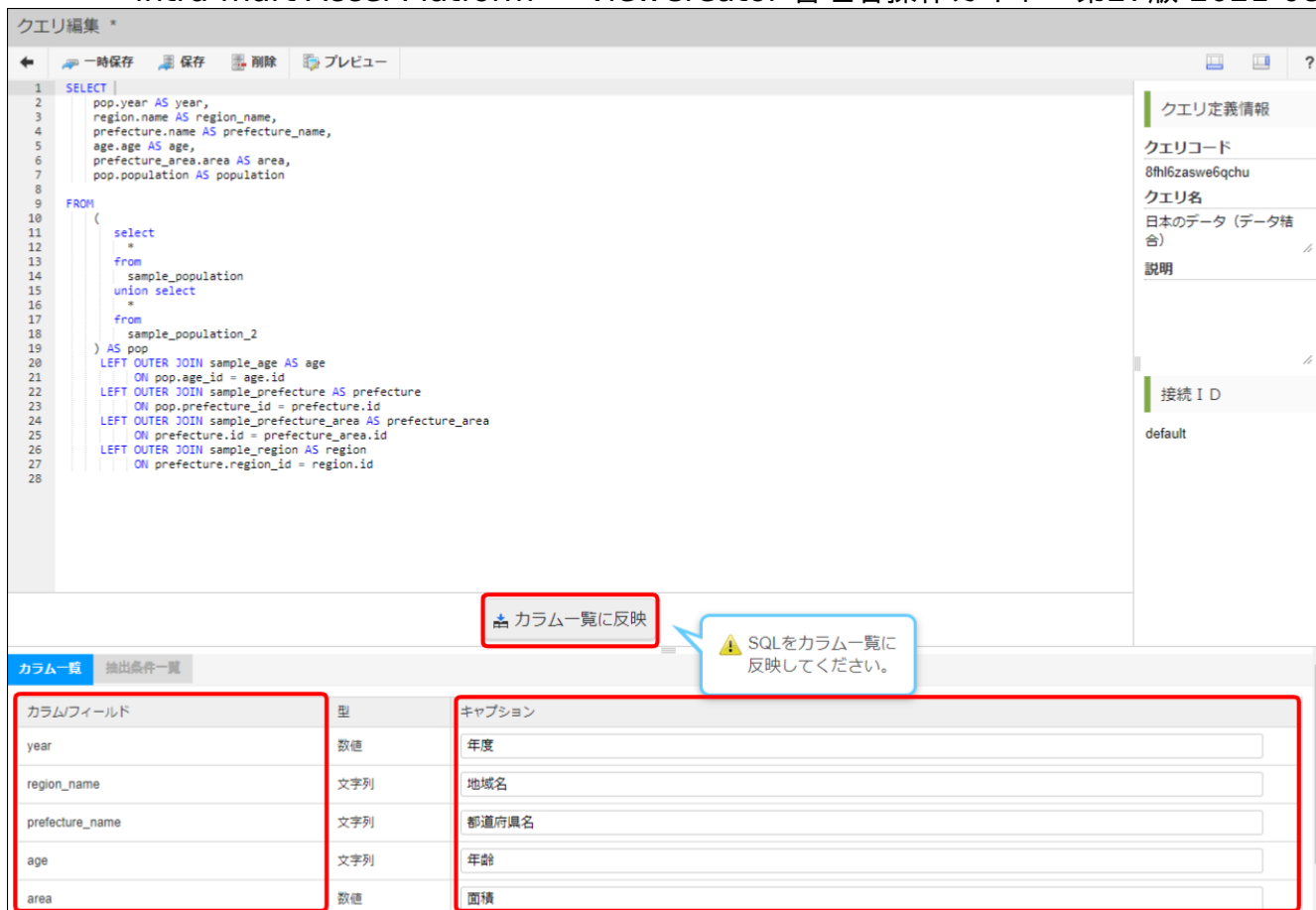
### コラム

SQLビルダではテナントデータベースかシェアドデータベースのみ利用可能です。  
外部データソースは利用できません。

- クエリのカラムを設定する部分です。  
「カラム一覧に反映」ボタンをクリックすることによって、エディタに記述したSQLからカラムを抽出し反映します。  
なお、SQLビルダでは抽出条件一覧は利用できません。

## カラムの追加

SQLを記述後、「カラム一覧に反映」をクリックするとSQLからカラム情報が抽出され、一覧に反映されます。



図：クエリ編集画面 - カラム追加

項目名	説明
カラム一覧に反映	エディタに記述したSQLからカラムを抽出し、一覧に反映します。 不正なSQLやSELECT文以外のSQLを記述した場合はカラム一覧に反映できません。
カラム/フィールド	クエリで利用するカラムです。 「カラム一覧に反映」をクリックすることによって追加されます。
キャプション	カラムに対するラベル名を設定します。 データ参照作成時に項目名の初期値としてセットされる他、計算式において変数名として利用できます。

## プレビュー

「プレビュー」を選択すると作成したクエリの実行結果を確認できます。

The 'Preview' window displays the following data:

年度	地域名	都道府県名	年齢	面積	人口
2015	九州	宮崎	0 - 14歳	6684.81	150
1990	東北	山形	15 - 64歳	7394.38	819
1970	九州	福岡	15 - 64歳	4843.11	2792
1985	関東	東京	65歳以上	2102.39	1056
2010	中部	福井	15 - 64歳	4189.22	491
1980	東北	山形	65歳以上	7394.38	147
1970	九州	佐賀	0 - 14歳	2439.54	215
2010	四国	愛媛	15 - 64歳	5677.03	865
2005	関東	東京	65歳以上	2102.39	2325
1970	中国	山口	65歳以上	6111.17	137
1995	関東	栃木	0 - 14歳	6408.28	339

**i コラム**  
 タイムスタンプ型のコラムは、プレビュー時はISO 8601 形式の日時フォーマットで表示されます。  
 例：2017-08-01T12:34:56.789Z

### クエリを保存する

画面上部の「保存」ボタンで作成したクエリを登録します。

The screenshot shows the 'クエリ編集' (Query Editor) interface. At the top, there are buttons for '一時保存' (Save Temporarily), '保存' (Save), '削除' (Delete), and 'プレビュー' (Preview). The '保存' button is highlighted with a red box. Below the buttons is a text area containing a SQL query. To the right of the query is a sidebar with 'クエリ定義情報' (Query Definition Information) and '接続ID' (Connection ID) sections. Below the query editor is a 'カラム一覧' (Column List) table.

カラムフィールド	型	キャプション
year	数値	年度
region_name	文字列	地域名
prefecture_name	文字列	都道府県名
age	文字列	年齢
area	数値	面積

図：クエリ編集画面 - 保存

クエリ定義情報の保存または更新ダイアログが表示されます。

The screenshot shows a '確認' (Confirmation) dialog box with the following fields:

- クエリコード\*: sample
- クエリ名\*: サンプル
- 説明: サンプルのクエリ定義です。

Buttons at the bottom: 決定 (OK), 取り消し (Cancel).

図：クエリ編集画面 - 保存ダイアログ

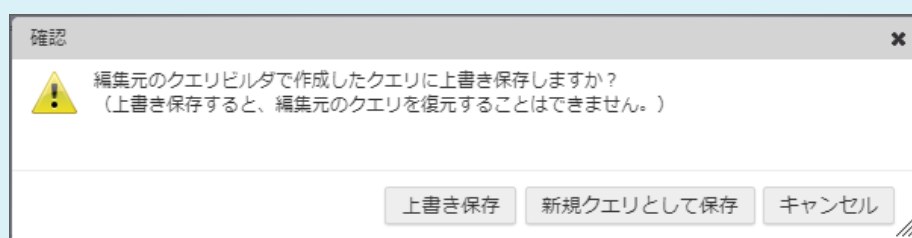
項目名	説明
-----	----

項目名	説明
クエリコード	クエリ定義を識別する一意のコードを設定します。 クエリコードは必須項目で、利用可能な文字は半角英数字とアンダースコアです。 新規作成時のみ入力可能で、編集はできません。
クエリ名	クエリ名を設定します。 必須項目です。
説明	クエリ定義の説明を設定できます。 任意項目です。

### コラム

クエリビルダで作成し保存したクエリを、SQLビルダに切り替えた場合、「保存」ボタンをクリックすると「上書き保存」か「新規クエリとして保存」が選択できます。

「新規クエリとして保存」を選択した場合は、クエリビルダで作成された既存のクエリは変更されず、新規クエリとして保存します。



### 注意

クエリビルダで作成し保存したクエリを、SQLビルダに切り替え、「上書き保存」を行った場合、クエリビルダに戻すことはできません。

また、SQLビルダで作成したクエリを、クエリビルダで編集することはできません。

### 注意

既存のクエリの編集は、管理者ユーザかそのクエリを作成したユーザのみが実行できます。

## データ参照の作成

クエリを元にしてデータ参照（データの見せ方）を作成します。

## 目次

- データ参照の種類の選択
- リスト集計の作成
  - 表示設定
  - アイコン画像の表示
  - 複数の表示条件設定（ヒートマップの作成）
  - サマリ集計の設定
  - リスト集計からグラフ集計を作成
- 作成したデータ参照の閲覧
- クロス集計
  - 集計方法とは
  - 小計行の表示
  - ソートについて
  - カラムの入れ替えと再表示
  - クロス集計の編集画面
- グラフ集計の作成
- 検索設定について
  - AND検索とOR検索（各項目を結合する演算子）について
  - リスト集計とサマリ集計で利用可能な機能
  - クロス集計で利用可能な機能
- 参照権の設定

## データ参照の種類の選択

1. 「サイトマップ」をクリックします。
2. 「ViewCreator」→「クエリー一覧」をクリックします。

作成する集計パターン（リスト集計、クロス集計、グラフ集計）を選択して、対応するアイコンをクリックします。

クエリー一覧

新規 コピー 削除 データ参照一覧へ

検索 クリア

データ参照作成	クエリー名	クエリーコード	接続ID	更新日	作成者	エクスポート
	system_log	5ib8yonru57ws	files	6月 12, 2014 10:58 午	system	
	transition_log	5ib8yony9uhzy	files	6月 12, 2014 10:58 午	system	
	ViewCreatorの設定テーブルを利	5i7ur3opljh2x	default	11月 6, 2013 2:40 午	tenant	
	SQL ViewCreatorの設定テーブルを利	5i7ur3opljh2xa	default	4月 27, 2021 3:45 午	imart	
	日本のデータ	5i7o4o3kli1pj	default	10月 29, 2013 2:26 午	tenant	
	SQL 日本のデータ (データ結合)	8fhl6zaswe6qchu	default	2月 6, 2020 3:28 午後	tenant	

1 ページ中 1 ページ目 50 6 件中 1 - 6 を表示

ファイル追加... 中断 削除

インポート

## アイコン 遷移先

リスト集計（サマリ集計）のデータ参照・新規作成画面に遷移します。



## アイコン 遷移先

 クロス集計のデータ参照・新規作成画面に遷移します。

 グラフ集計のデータ参照・新規作成画面に遷移します。

### コラム

データ参照もクエリと同様に、「エクスポート」アイコンでXMLファイルを出力できます。  
出力したデータは、データ参照一覧画面でインポートできます。

## リスト集計の作成

リスト集計作成アイコンをクリックすると次の画面に遷移します。

データ参照・編集 リスト/サマリ集計

← 削除 プレビュー

---

**データ参照設定**

データ参照コード *	5i7urzmrwxmufp4
データ参照名 *	地域 - 都道府県 - 年度 グループ化リスト(人口) ▶ 国際化データ
クエリ名	日本のデータ
集計パターン	リスト ▾
説明	<input type="text"/>
エクスポート機能の利用	<input checked="" type="checkbox"/> 利用する

**表示設定**

枠線	<input type="checkbox"/> 枠線を表示する
1行おきの背景色	<input type="text" value="#ffebd8"/>
表全体の幅	<input type="text" value="600"/> px
ヘッダ行の改行	<input type="checkbox"/> 改行する

カテゴリ	項目名	説明
データ参照設定	データ参照コード	半角英数字で入力してください。 ※一度登録したら変更できません。

**タイムゾーン設定**

タイムゾーン設定の表示  タイムゾーン設定を表示する

画面表示のデフォルト・タイムゾーン (デフォルト)ユーザが利用するタイムゾーン ▼

CSV出力のデフォルト・タイムゾーン (デフォルト)(GMT+00:00) UTC ▼

**カラム一覧**

計算式を追加

カラムの国際化項目の編集

カラム	タイプ	表示	フォーマット	ソート順	パラメータ名	表示設定
▲ ▼ 地域名(name) ▼	グループ化 ▼	<input checked="" type="checkbox"/>		▼		
▲ ▼ 都道府県名(name) ▼	グループ化 ▼	<input checked="" type="checkbox"/>		▼		
▲ ▼ 年度(year) ▼	グループ化 ▼	<input checked="" type="checkbox"/>	# 3桁区切り	▼		
▲ ▼ 年齢(age) ▼	▼	<input checked="" type="checkbox"/>		▼		
▲ ▼ 面積(area) ▼	平均 ▼	<input type="checkbox"/>	### 3桁区切り	▼		
▲ ▼ 人口(population) ▼	合計 ▼	<input checked="" type="checkbox"/>	### 3桁区切り	▼		
✖ ▲ ▼ キャプション 人口密度 式: 人口/面積 <input type="text"/> <input type="button" value="fx"/>		<input checked="" type="checkbox"/>	### 3桁区切り			<b>fx</b> ★

カテゴリ	項目名	説明
タイムゾーン設定	タイムゾーン設定の表示	データ参照一覧画面のタイムゾーン設定の表示/非表示を設定します。 表示する場合、データ参照一覧画面で画面表示、CSV出力のタイムゾーンを設定できます。
	画面表示のデフォルト・タイムゾーン	画面表示時に適用されるタイムゾーンを設定します。
	CSV出力のデフォルト・タイムゾーン	CSV出力時に適用されるタイムゾーンを設定します。
カラム一覧	上下アイコン	カラムの順番を入れ替えます。 一番上に設定されたカラムが、参照するとき一番左に表示されます。
	表示	列の表示/非表示を設定します。
	フォーマット	数値、日付、タイムスタンプ型のカラムのみ設定できます。 例) ###,###.## (数値) ⇒3桁区切り、小数点以下2位まで表示 yyyy/MM/dd HH:mm:ss (日付、タイムスタンプ)
	表示設定	表示上の設定を行う。 ポップアップ画面を表示します。

### コラム

「エクスポート機能の利用」はデフォルトでチェックON状態です。チェックOFF状態にすると、データ参照画面で以下のボタンを非表示状態にできます。

集計パターン	非表示
リスト集計、サマリ集計	「CSV出力」ボタン、「帳票出力」ボタン
クロス集計、グラフ集計	「帳票出力」ボタン

### コラム

フォーマットの設定は、データ参照を「CSV出力」する際にも適用されます。

## i コラム

- ソート順について

複数のカラムに対してソート順が設定された場合は、上から順に第1ソートキー、第2ソートキー、・・・と割り当てられます。下記の設定例では、地域名カラム（昇順）が第1ソートキー、都道府県名カラム（降順）が第2ソートキーです。

カラム	タイプ	表示	フォーマット	ソート順
地域名(name)		<input checked="" type="checkbox"/>		昇順
都道府県名(name)		<input checked="" type="checkbox"/>		降順
年齢(age)		<input checked="" type="checkbox"/>		
面積(area)		<input checked="" type="checkbox"/>	# 3桁区切り	
人口(population)		<input checked="" type="checkbox"/>	# 3桁区切り	

※ソート順は非表示カラムにも適用させることができます。

- 表示例

地域名	都道府県名	年齢	面積	人口
関東	栃木	65歳以上	6408	345
関東	栃木	0 - 14歳	6408	307
関東	栃木	15 - 64歳	6408	1352
関東	東京	15 - 64歳	2102	8686
関東	東京	0 - 14歳	2102	1421
関東	東京	65歳以上	2102	1910
関東	千葉	65歳以上	4996	837
関東	千葉	0 - 14歳	4996	843
関東	千葉	15 - 64歳	4996	4236
関東	神奈川	15 - 64歳	2416	6121
関東	神奈川	0 - 14歳	2416	1184
関東	神奈川	65歳以上	2416	1170
関東	埼玉	0 - 14歳	3767	1025

表示画面では、カラムのヘッダをクリックすることでソート順を切り替えることが可能です。

編集画面でのソート順の設定値に関係なく、最初のクリックで降順、2回目のクリックで昇順に並びます。

ヘッダをクリックされた列が第1ソートキーにセットされ、編集画面で設定されているソート順は第2ソートキー以下として利用されます。

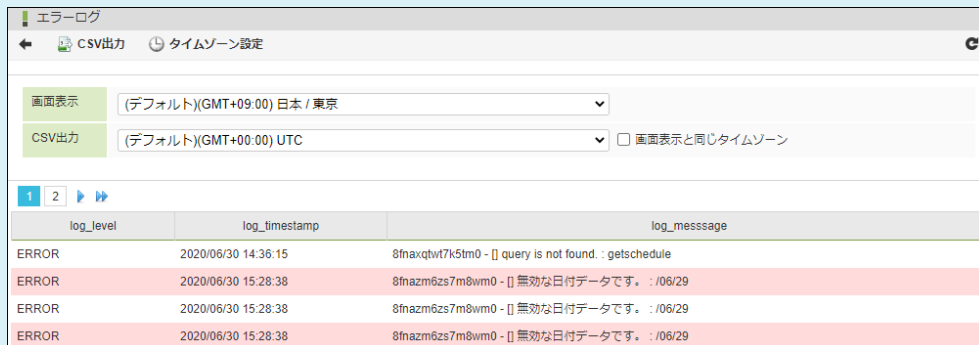
## コラム

データ参照編集、データ参照一覧画面に表示されるタイムゾーンは以下の優先順で表示されます。

画面表示：データ参照編集画面で登録されたタイムゾーン > 設定ファイル > ユーザのタイムゾーン

CSV出力：データ参照編集画面で登録されたタイムゾーン > 設定ファイル > システム・デフォルトのタイムゾーン  
タイムゾーン設定は2020 Summer(Zephyrine)から利用可能です。

### ■ タイムゾーン設定の表示例



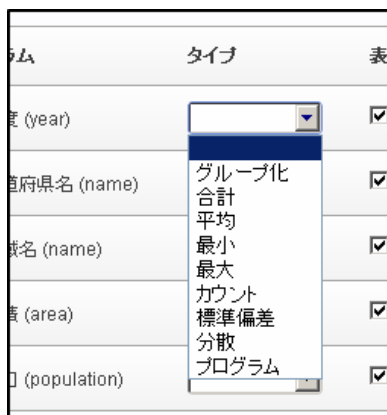
log_level	log_timestamp	log_message
ERROR	2020/06/30 14:36:15	8fnaxqwt7k5tm0 - [] query is not found. : getschedule
ERROR	2020/06/30 15:28:38	8fnazm6zs7m8wrm0 - [] 無効な日付データです。 : /06/29
ERROR	2020/06/30 15:28:38	8fnazm6zs7m8wrm0 - [] 無効な日付データです。 : /06/29
ERROR	2020/06/30 15:28:38	8fnazm6zs7m8wrm0 - [] 無効な日付データです。 : /06/29

## 注意

- 表示するデータ参照に日付型、タイムスタンプ型のカラムがない場合、ツールバーにタイムゾーン設定は表示されません。
- エクスポート機能の利用を利用しない場合、CSV出力のタイムゾーンを設定できません。
- 設定したタイムゾーンが適用されるのは、画面表示、CSV出力、検索条件のみです。クエリ、データ参照に設定した抽出条件には適用されません。

クエリで設定されたカラムが一覧で表示されるので、カラムごとに表示形式を設定していきます。

「タイプ」には集計関数および、グループ化、または任意のページへ遷移させるリンクの表示のいずれかを指定します。



カラム	タイプ	表
年 (year)		<input checked="" type="checkbox"/>
道府県名 (name)	グループ化	<input checked="" type="checkbox"/>
姓 (name)	合計	<input checked="" type="checkbox"/>
市 (area)	平均	<input checked="" type="checkbox"/>
人口 (population)	最小	<input checked="" type="checkbox"/>
	最大	<input checked="" type="checkbox"/>
	カウント	<input checked="" type="checkbox"/>
	標準偏差	<input checked="" type="checkbox"/>
	分散	<input checked="" type="checkbox"/>
	プログラム	<input checked="" type="checkbox"/>

### ■ グループ化

そのカラムを「GROUP BY」によってグループ化します。

グループ化したカラムはドリルダウンが可能です。

- 集計関数 (合計、平均、最小、最大、カウント、標準偏差、分散)

全レコードを対象とした計算結果と、グループ化が指定されている場合は各グループごとに小計を表示します。

### ■ プログラム

任意のプログラム (ページ) へ連携します。

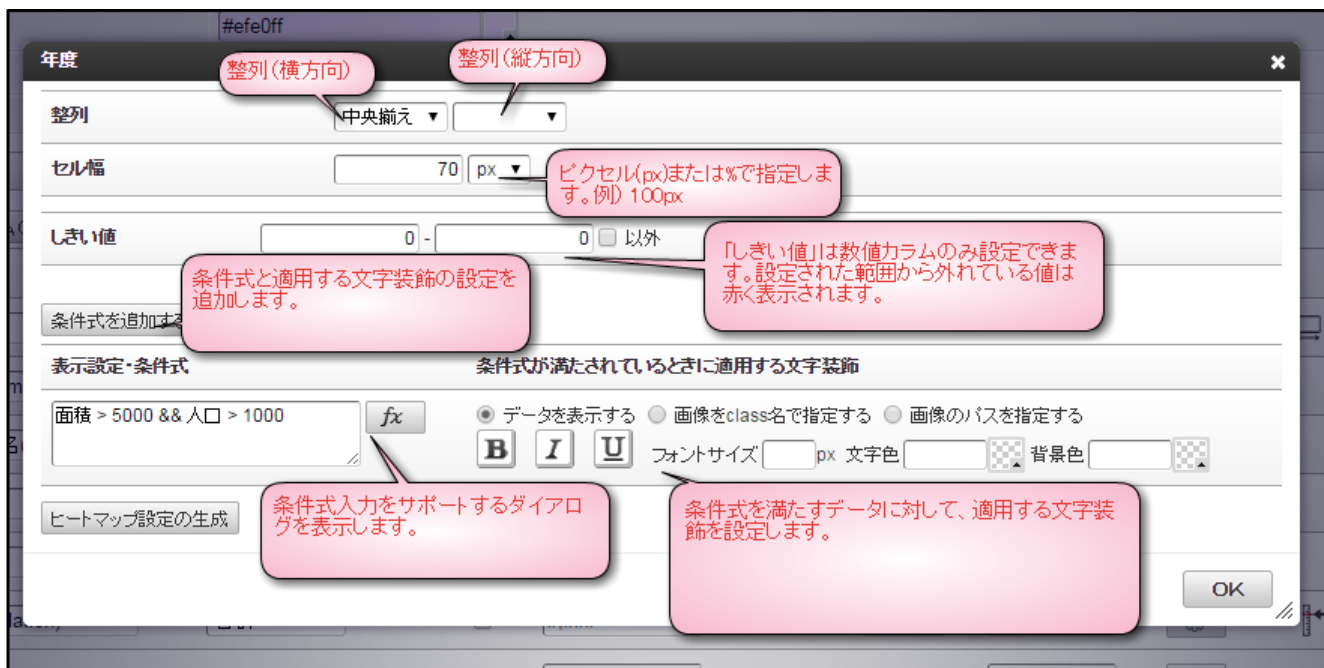
詳細は応用編「[外部ページへの連携 \(リスト/サマリ集計のみ\)](#)」を参照してください。

## 表示設定

表示設定ボタンアイコンをクリックすると、下記のようなポップアップが表示されます。

**i** コラム

縦方向の整列設定は2014 Winter(Iceberg)から利用可能です。



しきい値の設定と条件式の設定が両方満たされた場合は、条件式による設定が優先されます。



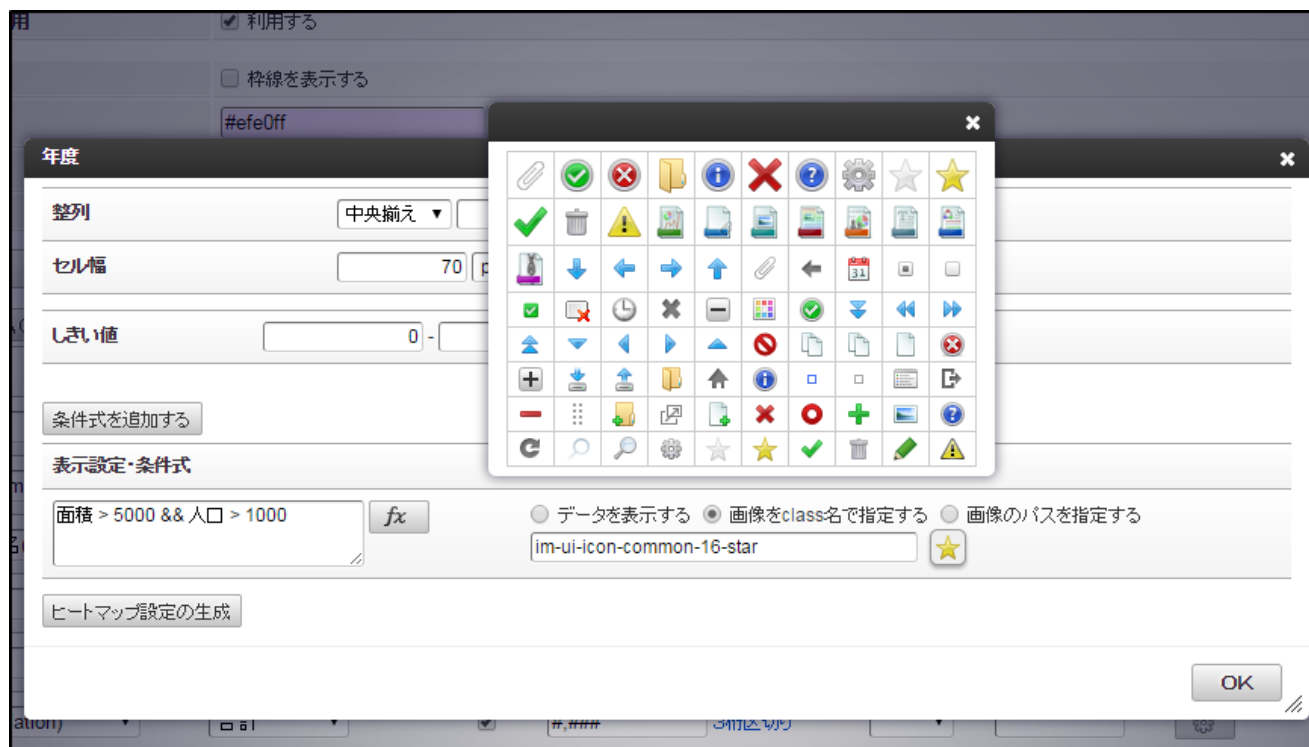
条件式は手入力することも可能ですが、式の作成を補助するダイアログを利用することも可能です。

**!** 注意

条件式で設定された式が真偽値(TRUE/FALSE)以外のデータを返した場合は、すべてFALSEと判定されます。

## アイコン画像の表示

文字装飾ではなく代わりにアイコンを表示させることも可能です。



アイコン画像は一覧から選ぶだけでなく、class名や画像のパス名を直接指定できます。

**i** コラム

- 設定例

例えばデータ参照名だけ入力して、それ以外は標準設定のままプレビューした結果、このように表示されたとします。

intra-mart® Top ▾ テナント管理 ▾ ViewCreator ▾ サン

リストサンプル

CSV出力

◀◀ 1 2 3 4 5 6 7 8 9 10 ▶▶

年度	地域名	都道府県名	面積	人口
1985	関東	茨城	6096	279
1980	関東	茨城	6096	236
1975	関東	茨城	6096	196
1970	関東	茨城	6096	169
2000	関東	茨城	6096	2030
1995	関東	茨城	6096	2030
1990	関東	茨城	6096	1944
1985	関東	茨城	6096	1819
1980	関東	茨城	6096	1692
1975	関東	茨城	6096	1565
1970	関東	茨城	6096	1440
2000	関東	茨城	6096	459
1995	関東	茨城	6096	505
1990	関東	茨城	6096	559
1985	関東	茨城	6096	628
1980	関東	茨城	6096	628
1975	関東	茨城	6096	580
1970	関東	茨城	6096	534
2000	関東	栃木	6408	345
1995	関東	栃木	6408	293
1990	関東	栃木	6408	239

プレビューを閉じて、下記のように設定します。

カラム	タイプ	表示	フォーマット	ソート順	パラメータ名	表示設定
▲▼ 年度(year)	グループ化	<input checked="" type="checkbox"/>	# 3桁区切り	降順		
▲▼ 地域名(name)	グループ化	<input checked="" type="checkbox"/>				
▲▼ 都道府県名(name)	グループ化	<input checked="" type="checkbox"/>				
▲▼ 面積(area)		<input checked="" type="checkbox"/>	### 3桁区切り			
▲▼ 人口(population)	平均	<input checked="" type="checkbox"/>	### 3桁区切り			

右寄せ  
しきい値

しきい値 200 - 1000

このように表示されます。

リストサンプル

CSV出力

1

グループ化した列は展開することが可能です。

集計関数を設定すると小計や合計が表示されます。また、しきい値の範囲外の値は赤く表示されます。

トップ階層へ戻ります。

年度	地域名	都道府県名	面積	人口(平均)
1985	関東	+ 茨城	-	909
		+ 栃木	-	622
		+ 埼玉	-	1,954
		+ 千葉	-	1,715
		+ 神奈川	-	2,476
		+ 群馬	-	641
		+ 東京	-	3,940
				1,751

### i コラム

枠線の表示/非表示や背景色を設定の設定や、表全体の幅を指定することも可能です。

説明

枠線  枠線を表示する

1行おきの背景色 #ffebd8

表全体の幅 600px

ヘッダー行の改行  改行する

エクスポート機能の利用  利用する

コラム一覧

実行結果

リストサンプル

CSV出力

1

年度	地域名	都道府県名	面積(平均)	人口(合計)
1985	関東	+ 千葉	4,996	5,146
		+ 埼玉	3,767	5,861
		+ 東京	2,102	11,819
		+ 栃木	6,408	1,866
		+ 神奈川	2,416	7,429
		+ 群馬	6,363	1,922
		+ 茨城	6,096	2,726
			4,593	36,769

### 複数の表示条件設定（ヒートマップの作成）

「条件式を追加する」ボタンをクリックすると、表示条件を複数個設定できます。複数の条件に合致した場合は、一覧上でより下に設定されている条件が優先されて適用されます。これを利用して、数値の大きさに応じて背景色や文字色を段階的に変える等の設定が可能です。



## コラム

複数の表示条件設定は2014 Summer(Honoka)から利用可能です。

**年度**

整列 中央揃え ▼

セル幅 70 px ▼

しきい値 0 - 0  以外

表示設定・条件式	条件式が満たされているときに適用する文字装飾
年度 > 0 && 3200 >= 年度 <span style="float: right; border: 1px solid gray; padding: 2px;">fx</span>	<input checked="" type="radio"/> データを表示する <input type="radio"/> 画像をclass名で指定する <input type="radio"/> 画像のパスを指定する <input checked="" type="checkbox"/> <b>B</b> <input type="checkbox"/> <b>I</b> <input type="checkbox"/> <b>U</b> フォントサイズ <input type="text"/> px 文字色 <input type="text"/> 背景色 <input style="background-color: #ffffff;" type="text"/> #ffffff
年度 > 3200 && 6400 >= 年度 <span style="float: right; border: 1px solid gray; padding: 2px;">fx</span>	<input checked="" type="radio"/> データを表示する <input type="radio"/> 画像をclass名で指定する <input type="radio"/> 画像のパスを指定する <input checked="" type="checkbox"/> <b>B</b> <input type="checkbox"/> <b>I</b> <input type="checkbox"/> <b>U</b> フォントサイズ <input type="text"/> px 文字色 <input type="text"/> 背景色 <input style="background-color: #ccccff;" type="text"/> #ccccff
年度 > 6400 && 9600 >= 年度 <span style="float: right; border: 1px solid gray; padding: 2px;">fx</span>	<input checked="" type="radio"/> データを表示する <input type="radio"/> 画像をclass名で指定する <input type="radio"/> 画像のパスを指定する <input checked="" type="checkbox"/> <b>B</b> <input type="checkbox"/> <b>I</b> <input type="checkbox"/> <b>U</b> フォントサイズ <input type="text"/> px 文字色 <input type="text"/> 背景色 <input style="background-color: #9999ff;" type="text"/> #9999ff
年度 > 9600 && 12800 >= 年度 <span style="float: right; border: 1px solid gray; padding: 2px;">fx</span>	<input checked="" type="radio"/> データを表示する <input type="radio"/> 画像をclass名で指定する <input type="radio"/> 画像のパスを指定する <input checked="" type="checkbox"/> <b>B</b> <input type="checkbox"/> <b>I</b> <input type="checkbox"/> <b>U</b> フォントサイズ <input type="text"/> px 文字色 <input type="text"/> 背景色 <input style="background-color: #6666ff;" type="text"/> #6666ff
年度 > 12800 && 16000 >= 年度 <span style="float: right; border: 1px solid gray; padding: 2px;">fx</span>	<input checked="" type="radio"/> データを表示する <input type="radio"/> 画像をclass名で指定する <input type="radio"/> 画像のパスを指定する <input checked="" type="checkbox"/> <b>B</b> <input type="checkbox"/> <b>I</b> <input type="checkbox"/> <b>U</b> フォントサイズ <input type="text"/> px 文字色 <input type="text"/> 背景色 <input style="background-color: #3333ff;" type="text"/> #3333ff

## コラム

追加した条件を削除したい場合は、式の入力欄を空にしてデータ参照を登録してください。

「ヒートマップ設定の生成」機能を利用すると、開始値と終了値の入力だけで背景色をグラデーション的に表示する設定を自動的に生成できます。

#efe0ff

年度

整列 中央揃え

セル幅 70 px

しきい値 0 - 0  以外

条件式を追加する

表示設定・条件式 条件式が満たされているときに適用する文字装飾

面積 > 5000 && 人口 > 1000 fx  データを表示する  画像をclass名で指定する  画像のパスを指定する

im-ui-icon-common-16-star ★

ヒートマップ設定の生成

ステップ数 5

開始値 0

開始カラー #ffffff

終了値 16000

終了カラー #0000ff

生成する

## サマリ集計の設定

リスト集計の従属集計パターンとして、サマリ集計が用意されています。

クエリ名	日本のデータ
集計パターン	リスト リスト サマリ
説明	

サマリ集計が選択された場合は、グループ化カラムをドリルダウン形式で1階層ずつ展開するのではなく設定されたグループ化列をまとめてグルーピングします。

また、抽出条件の設定を行うことで集合関数を使用できます。

## リスト集計からグラフ集計を作成

グラフ集計はクエリから作成するだけでなく、リスト（サマリ）集計から作成することも可能です。

データ参照一覧画面において、リスト（サマリ）集計の場合のみ「データ参照作成」列にグラフ集計作成アイコンが表示されます。

### コラム

- クエリからグラフを作成する
- リスト（サマリ）集計からグラフを作成する

いずれの場合においてもグラフ集計作成画面で設定可能な項目は同一です。

データ参照一覧に追加されます。

<input type="checkbox"/>	編集	データ参照作成	データ参照名	更新日	作成者	エクスポート
<input type="checkbox"/>			[ポータル用]トランジションログ・ページ別グループ化リスト	2020/02/26 14:30	tenant	
<input type="checkbox"/>			2000年度版・地域・都道府県 サマリ集計	2020/02/26 14:30	tenant	
<input type="checkbox"/>			2000年度版・地域・都道府県 サマリ集計/面積順	2013/10/31 4:20	tenant	
<input type="checkbox"/>			2010年度版・地域・都道府県 サマリ集計	2020/02/06 15:47	tenant	
<input type="checkbox"/>			2010年度版・地域・都道府県 サマリ集計/面積順	2020/02/06 15:38	tenant	

項目名	説明
編集	アイコンからデータ参照の変更画面に遷移します。 データ参照作成者、または管理者のみ実行できます。
データ参照作成	リスト集計から別の集計（グラフ集計）を作成できます。アイコンからグラフ集計のデータ参照・新規作成画面に遷移します。 データ参照作成者、または管理者のみ実行できます。
データ参照名	リンクから該当のデータ参照を使った閲覧画面に遷移します。

データ参照名のリンクをクリックすると、データの閲覧ができます。

**intra-mart** Top ▾ テナント管理 ▾ ViewCreator ▾ サンプル ▾

リストサンプル

← CSV出力 🔍 検索

1

年度	地域名	都道府県名	面積	人口(平均)
+ 2000	-	-	-	899
+ 1995	-	-	-	890
+ 1990	-	-	-	874
+ 1985	-	-	-	858
+ 1980	-	-	-	830
+ 1975	-	-	-	794
+ 1970	-	-	-	742
				<b>841</b>

### コラム

出力されたCSVファイルをMicrosoft Excelで開く場合はCSVファイルをダブルクリック等で開くのではなく「外部データの取り込み」機能を利用してください。

例) Microsoft Excel 2013の場合は、以下の操作でBOMなしUTF-8のテキストファイルを読み込むことができます。  
「データ」タブをクリックします。  
「外部データの取り込み・テキストファイル」をクリックします。  
読み込むcsvファイルを選択します

**!** 注意

2016 Spring(Maxima) より、「[リスト集計におけるCSV出力の設定](#)」においてBOMを付与できるようになりました。そのため、出力されたCSVファイルのファイルエンコーディングがUTF-8の場合でも、Microsoft Excelで開く際にダブルクリック等で開ける可能性があります。

**i** コラム

CSVファイルのヘッダには、参照元クエリのカラムで設定されているキャプションが出力されます。

クロス集計

クロス集計とは、2つの軸でデータを横断的に分析することを可能にする集計方法です。

※下記はクロス集計の表示画面です。編集画面については後述します。

**i** コラム

クロス集計の表のデザインは2019 Spring(Violette)で一新されています。

年度・都道府県別人口 - クロス集計

← 検索 帳票出力

**入れ替え可能カラム一覧**

**列見出し**  
年度 ソート指定無し ▼

**行見出し**  
地域名 見出し名 ▼ ソート指定無し ▼  
都道府県名 見出し名 ▼ ソート指定無し ▼

**値**  
人口 合計 ▼ 数値フォーマット

表示/非表示  合計  平均  最小  最大  カウント

各項目を結合する演算子  AND  OR

地域名  すべて選択  
 北海道  東北  関東  中部  近畿  中国  四国  九州

人口 [ ] - [ ]

年度 1990 - 2000

表示 クリア

		1990	1995	2000	合計	平均
北海道	北海道	5,634	5,687	5,657	16,978	1,886.44
	小計				16,978	1,886.44
東北	青森	1,482	1,480	1,475	4,437	493
	岩手	1,417	1,419	1,415	4,251	472.33
	宮城	2,241	2,328	2,365	6,934	770.44
	秋田	1,228	1,214	1,189	3,631	403.44
	山形	1,258	1,257	1,244	3,759	417.67
	福島	2,102	2,134	2,127	6,363	707
	小計				29,375	543.98

上記の例では、縦に見れば年度ごと、横に見れば都道府県ごとの人口データを表示しています。

クロス集計には、次の3種類のカラムを設定できます。

- 列見出し

- 行見出し
- 値

列見出しと行見出しには「ソート順」、値に関しては「集計方法」を選択できます。

「ソート指定無し」が選択された場合は、データベースから取得されたレコード順に従って見出しの作成が行われます。



#### 注意

2015 Winter(Lydia)以降では、1つのカラムを複数個所に設定することはできません。  
例えば上記の例で、「年度」カラムを「行見出し」と「値」の両方に設定する場合はクエリ作成時に「年度」カラムを2つ作成してください。

## 集計方法とは

例)

年度	都道府県名	人口
1970	千葉	810
1970	千葉	200
1970	千葉	50
1970	東京	1500
1970	東京	350

このようなデータであれば、年度が“1970”で都道府県名が“千葉”のレコードは3つあります。

このとき、その3レコードの人口データ(810, 200, 50)をどのような方法で1つに纏めるのかを設定します。

⇒ 合計、平均、最大、最小、カウント、標準偏差、分散

\*カウントを選択した場合は値に関係なくレコード数が集計結果として表示されます。

「表示/非表示」の項目で表示されている、合計、平均、最小、最大、カウントのチェックボックスは、列ごと、行ごと、トータルの集計結果を表示するかどうかを設定できます。

## 小計行の表示

「行見出し」に複数のカラムが設定されている場合（カテゴリ化されている場合）は、「表示/非表示」の項目についてカテゴリ単位で小計を表示します。



#### コラム

小計行は 2019 Spring(Violette)より、表示されます。



#### コラム

「値」に複数のカラムが設定されている場合は、設定画面上で上にセットされたカラムのデータを小計行に表示します。

## ソートについて

「列見出し」と「行見出し」には、それぞれソート順を指定できます。

ソートは「見出し名(ラベル)」を利用して行います。

「行見出し」については、集計結果(合計、平均、最小、最大、カウント)を利用したソートを行うことも可能です。

「行見出し」に複数のカラムが設定されている場合（カテゴリ化されている場合）は、カテゴリ単位で行われた小計の値でソートします。



#### コラム

「行見出し」の集計結果を利用したソート機能は 2019 Spring(Violette)より、利用可能です。



#### 注意

「表示/非表示」がチェックされていない集計結果でソートすることはできません。

## カラムの入れ替えと再表示

カラムの入れ替えはドラッグ&ドロップで行うことができます。

「入れ替え可能カラム一覧」に配置されたカラムは、クロス集計の計算から除外されます。

「表示」ボタンをクリックすると、再計算が実行されます。

年度・都道府県別人口 - クロス集計

← 検索 帳票出力

<b>入れ替え可能カラム一覧</b> <div style="border: 1px solid red; padding: 2px; margin-bottom: 2px;">地域名</div> <div style="border: 1px solid gray; padding: 2px;">都道府県名</div>	<b>列見出し</b> 年度 <span style="border: 1px solid gray; padding: 2px;">ソート指定無し ▼</span>
<b>行見出し</b> 	<b>値</b> 人口 <span style="border: 1px solid gray; padding: 2px;">合計 ▼</span> 数値フォーマット <span style="border: 1px solid gray; padding: 2px;"> </span>

表示/非表示  合計  平均  最小  最大  カウント

各項目を結合する演算子  AND  OR

地域名  すべて選択  
 北海道  東北  関東  中部  近畿  中国  四国  九州

人口   -  

年度   -  

表示 クリア

	1970	1975	1980	1985	1990	1995	2000	合計	平均
合計	104,665	111,896	116,985	121,017	123,286	125,436	126,699	829,984	
平均	742.3	793.59	829.68	858.28	874.37	889.62	898.57		840.92

## クロス集計の編集画面

クロス集計の編集画面では、表示画面とほぼ同じようにカラムの設定を行うことができます。

ただし、表示画面と編集画面では以下の違いがあります。

- 表示画面で行ったカラム設定は保存されません。
- 編集画面では「カラム一覧」が表示されます。

表示設定

カラムの国際化項目の編集

<b>カラム一覧</b> <div style="border: 1px solid red; padding: 2px; margin-bottom: 2px;">年齢</div> <div style="border: 1px solid red; padding: 2px;">面積</div>	<b>入れ替え可能カラム一覧</b> 	<b>列見出し</b> 年度 <span style="border: 1px solid gray; padding: 2px;">ソート指定無し ▼</span>
	<b>行見出し</b> 地域名 <span style="border: 1px solid gray; padding: 2px;">見出し名 ▼</span> <span style="border: 1px solid gray; padding: 2px;">ソート指定無し ▼</span> 都道府県名 <span style="border: 1px solid gray; padding: 2px;">見出し名 ▼</span> <span style="border: 1px solid gray; padding: 2px;">ソート指定無し ▼</span>	<b>値</b> 人口 <span style="border: 1px solid gray; padding: 2px;">合計 ▼</span> 数値フォーマット <span style="border: 1px solid gray; padding: 2px;"> </span>

表示/非表示  合計  平均  最小  最大  カウント

数値フォーマット #,###,## 3桁区切り

カラム一覧には、クエリで設定されているカラムがすべて表示されます。

**i** コラム

「数値フォーマット」は「値」として設定されたカラムと、「表示/非表示」の下に表示されている入力項目の2箇所を設定できます。  
 後者の設定は、前者が未設定のときにのみ利用されます。

設定値として利用可能な文字列はどちらも同じです。  
 ※java.text.DecimalFormatで利用可能なフォーマット文字列を設定できます。  
 例)  
 3桁区切りにする場合：#,###  
 小数点以下第2位まで表示する場合：#.##

**i** コラム

クロス集計とグラフ集計にはページングが無いため、「1ページに表示する行数」で設定された値が集計対象とする最大取得レコード数です。  
 デフォルト値である0（ゼロ）です。  
 無制限にする場合は、0（ゼロ）を設定します。

「行見出し」、「列見出し」、「値」には複数のカラムを配置することも可能です。

年度・都道府県別人口 - クロス集計

← 検索 帳票出力

**入れ替え可能カラム一覧**

面積

**列見出し**

年度 ソート指定無し ▼

年齢 ソート指定無し ▼

**行見出し**

地域名 見出し名 ▼ ソート指定無し ▼

都道府県名 見出し名 ▼ ソート指定無し ▼

**値**

人口 合計 ▼ 数値フォーマット

表示/非表示  合計  平均  最小  最大  カウント

各項目を結合する演算子  AND  OR

地域名  すべて選択

北海道  東北  関東  中部  近畿  中国  四国  九州

人口  -

年度 1990 - 2000

表示 クリア

		1990			1995			2000			合計	平均
		0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上		
北海道	北海道	1,034	3,925	675	899	3,943	845	792	3,833	1,032	16,978	1,886.44
	小計										16,978	1,886.44
東北	青森	289	1,001	192	252	991	237	223	965	287	4,437	493
	岩手	270	941	206	239	925	255	212	899	304	4,251	472.33
	宮城	439	1,535	267	394	1,596	338	354	1,602	409	6,934	770.44
	秋田	220	816	192	189	787	238	163	746	280	3,631	403.44
	山形	234	819	205	209	799	249	186	772	286	3,759	417.67
	福島	422	1,378	302	382	1,380	372	341	1,354	432	6,363	707
	小計										29,375	543.98

**!** 注意

クロス集計では、対象となる全てのデータをメモリ上に展開して計算が実行されます。  
 データ量や表の大きさ（行見出しと列見出しの項目数に依存します）が大きすぎる場合は、エラーが発生する可能性があります。

グラフ集計において、グラフを描画するために利用するライブラリを選択できます。

グラフ描画ライブラリはデータ参照・編集画面の「グラフ描画形式」で切り替えが行えます。

intra-mart Accel Platform標準では以下の2種類の「グラフ描画形式」を提供しています。

## JFreeChart

データ参照・編集画面の「グラフ描画形式」で「JFreeChart」を選択した場合の表示形式です。

JFreeChartでは4種類のグラフを作成できます。

### コラム

URL (2019年6月現在)

JFreeChart : <http://www.jfree.org/jfreechart/>

## 棒グラフ

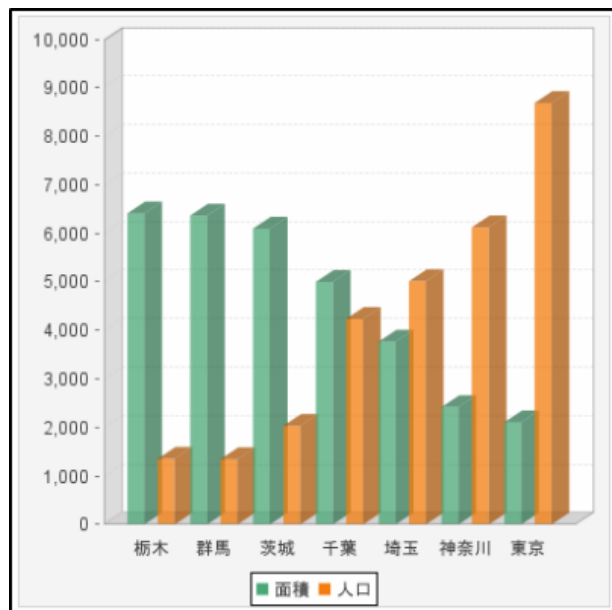


図 棒グラフィイメージ

## 折れ線グラフ

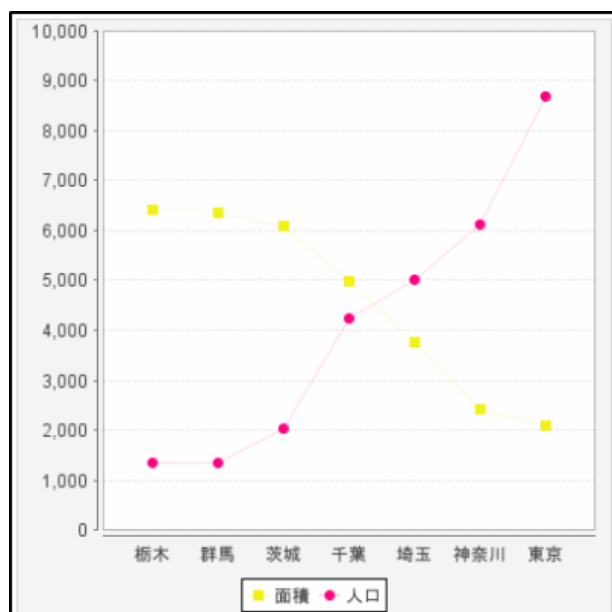


図 折れ線グラフィイメージ

## レーダーチャート



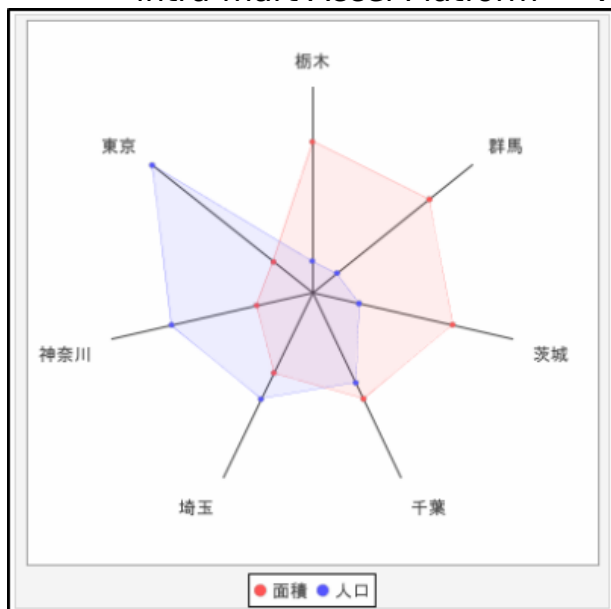


図 レーダーチャートイメージ

円グラフ

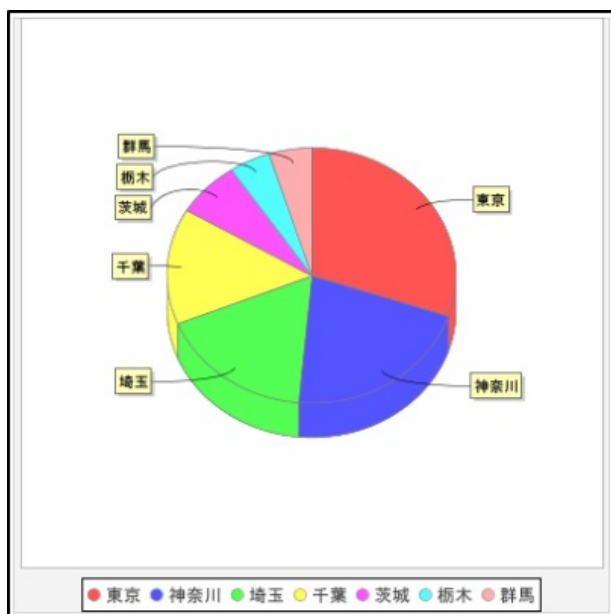


図 円グラフィメージ

グラフ設定

グラフ表示時のレイアウトを設定します。

グラフ設定	
グラフ描画形式*	JFreeChart ▼
グラフタイプ*	棒グラフ ▼
表示タイプ	<input type="radio"/> 水平表示 <input checked="" type="radio"/> 立体表示
グラフの大きさ(縦幅)	500
グラフの大きさ(横幅)	500
目盛りの最大値	15000
目盛りの最小値	0
1目盛りあたりの数	1000
100分率表示	<input checked="" type="checkbox"/>
リストの表示	<input checked="" type="checkbox"/>

## 設定項目

項目名	説明
グラフ描画形式	グラフの描画形式を選択します。 intra-mart Accel Platform標準では「JFreeChart」、「Highcharts」2種類のグラフ描画形式を提供しています。
グラフタイプ	グラフの種類を選択します。 JFreeChartでは「棒グラフ」、「折れ線グラフ」、「レーダーチャート」、「円グラフ」4種類のグラフを提供しています。
表示タイプ	水平表示（2D描画）、または立体表示（3D描画）を選択します。
グラフの大きさ（縦幅）	グラフの表示領域（縦）を設定します。
グラフの大きさ（横幅）	グラフの表示領域（横）を設定します。
目盛りの最大値	グラフ内のデータの中で最も大きい数値を設定します。 目盛りの最大値 = 目盛りの最小値の場合、グラフの上限値と下限値は自動計算されます。
目盛りの最小値	グラフ内のデータの中で最も小さい数値を設定します。 目盛りの最大値 = 目盛りの最小値の場合、グラフの上限値と下限値は自動計算されます。
1目盛りあたりの数	設定された値ごとに数字付きの目盛りが設定されます。 0以下の値が設定された場合は自動計算されます。
100分率表示	円グラフを表示した場合に、値カラムのデータの100分率をグラフ上に表示します。
リストの表示	グラフの数値データを表で添えるかどうかを設定します。

 コラム

- 目盛りの最大値、目盛りの最小値、1目盛りあたりの数は2014 Winter(Iceberg)から利用可能です。
- 円グラフ、レーダーチャートの場合、目盛りの最大値、目盛りの最小値、1目盛りあたりの数は設定できません。
- 円グラフ以外の場合、100分率表示は設定できません。
- 設定できない項目は非活性状態に変わります。

## コラム設定

グラフデータとして使用するデータが格納されているコラムを選択します。  
クエリに追加されたコラムの内、数値型のコラムのみが表示されます。

 注意

グラフ集計を作成するためには、最低1つ以上の数値コラムが必要です。

### コラム設定

コラムの国際化項目の編集

キャプションコラム	地域名(name) ▼	
凡例コードコラム	都道府県名(name) ▼	
凡例ラベルコラム	都道府県名(name) ▼	
値コラム	面積(area) ▼	

	表示	コラム	カラー
▲ ▼	<input checked="" type="checkbox"/>	年度	#aaffaa
▲ ▼	<input checked="" type="checkbox"/>	面積	#aaffff
▲ ▼	<input checked="" type="checkbox"/>	人口	#ffaaff

## 設定項目

項目名	説明
カラムの国際化項目の編集	カラムの国際化項目の編集ダイアログが表示されます。 詳細は、「 <a href="#">国際化とヘッダ・フッタ</a> 」を参照してください。
キャプションカラム	グラフのキャプションラベルとして使用するデータが格納されているカラムを選択します。
凡例コードカラム	詳細は、「 <a href="#">グラフ集計で「凡例」の表示方法を変更する</a> 」を参照してください。
凡例ラベルカラム	詳細は、「 <a href="#">グラフ集計で「凡例」の表示方法を変更する</a> 」を参照してください。
値カラム	円グラフを表示した場合の実データとして利用するカラムを選択します。
△▽アイコン	クリックで上下の順序を変更できます。 円グラフの場合、表示されません。
表示	選択したカラムがグラフ描画時に表示されます。 円グラフの場合、表示されません。
カラム	グラフを表示した場合の実データとして利用するカラムです。 円グラフの場合、表示されません。
カラー	グラフ描画する際の色彩を設定します。 円グラフの場合、表示されません。

**i** コラム

- 円グラフ、レーダーチャートの場合、凡例コードカラム、凡例ラベルカラムは設定できません。
- 円グラフ以外の場合、値カラムは設定できません。
- 設定できない項目は非活性状態に変わります。

## Highcharts

データ参照・編集画面の「グラフ描画形式」で「Highcharts」を選択した場合の表示形式です。  
Highchartsでは5種類のグラフを組み合わせで作成できます。

**i** コラム

URL (2019年6月現在)

Highcharts : <https://www.highcharts.com/>

## グラフ

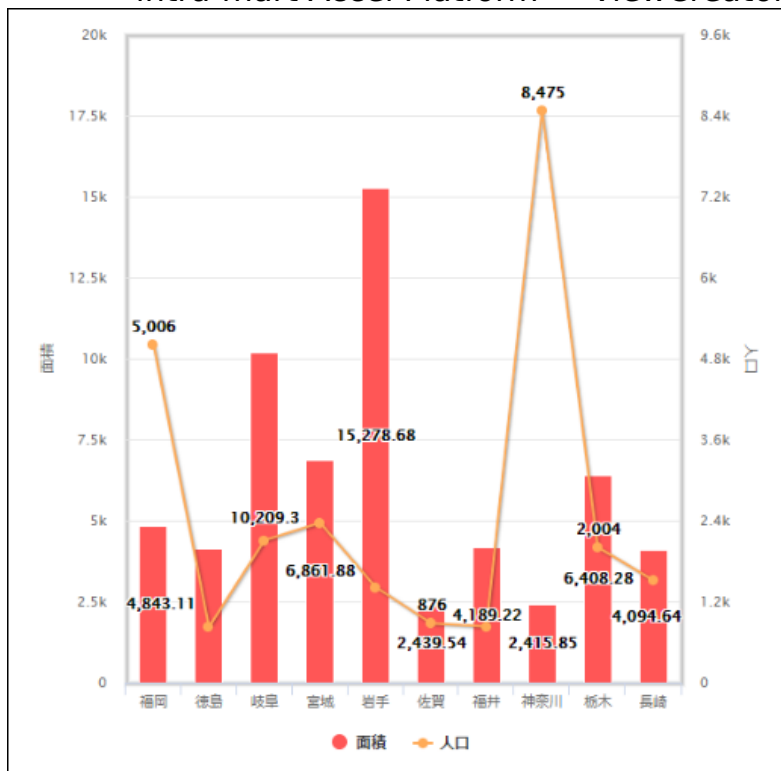


図 グラフ組み合わせイメージ

棒グラフ

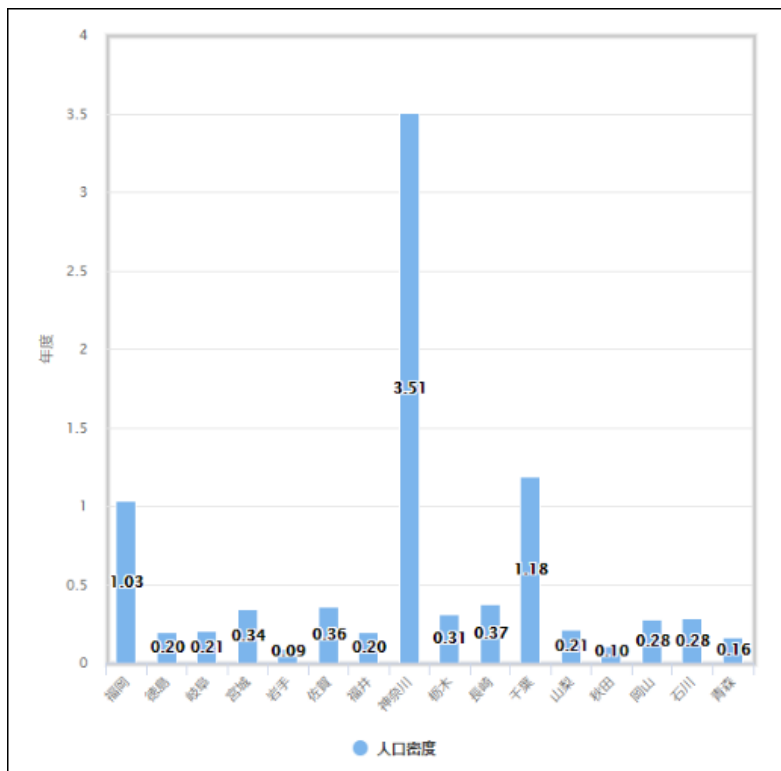


図 棒グラフィイメージ

折れ線グラフ

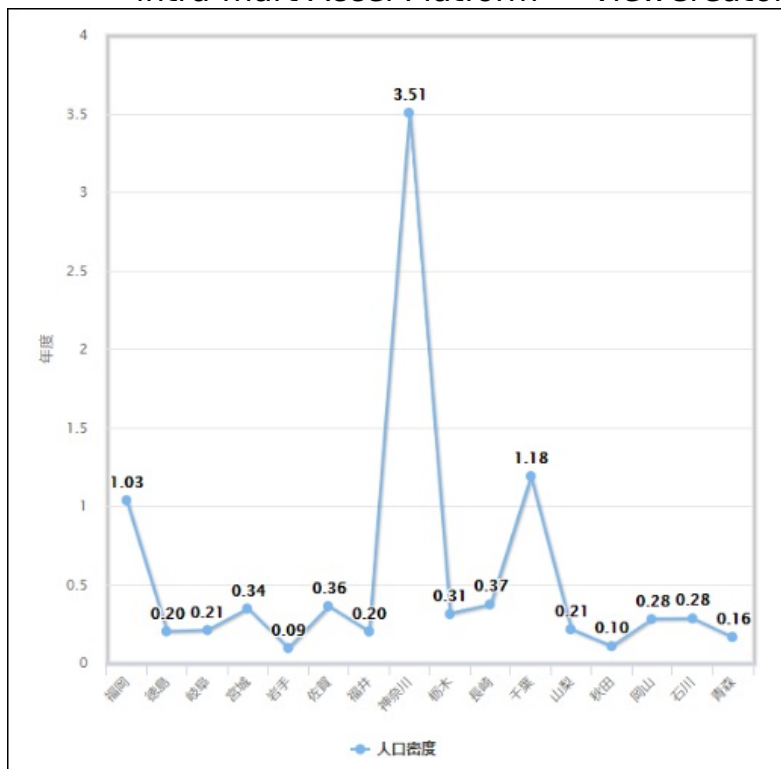


図 折れ線グラフィイメージ

レーダーチャート

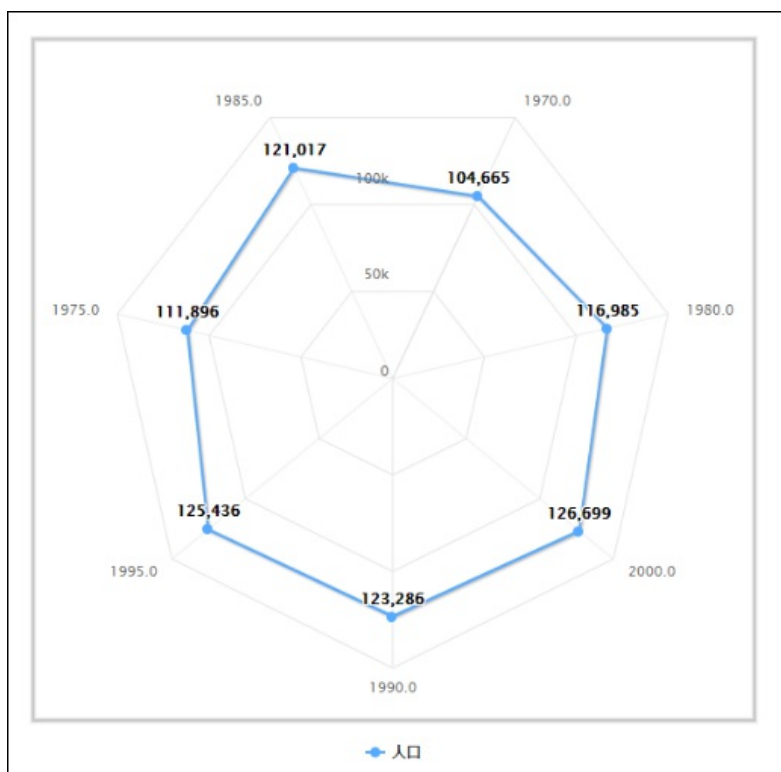


図 レーダーチャートイメージ

円グラフ

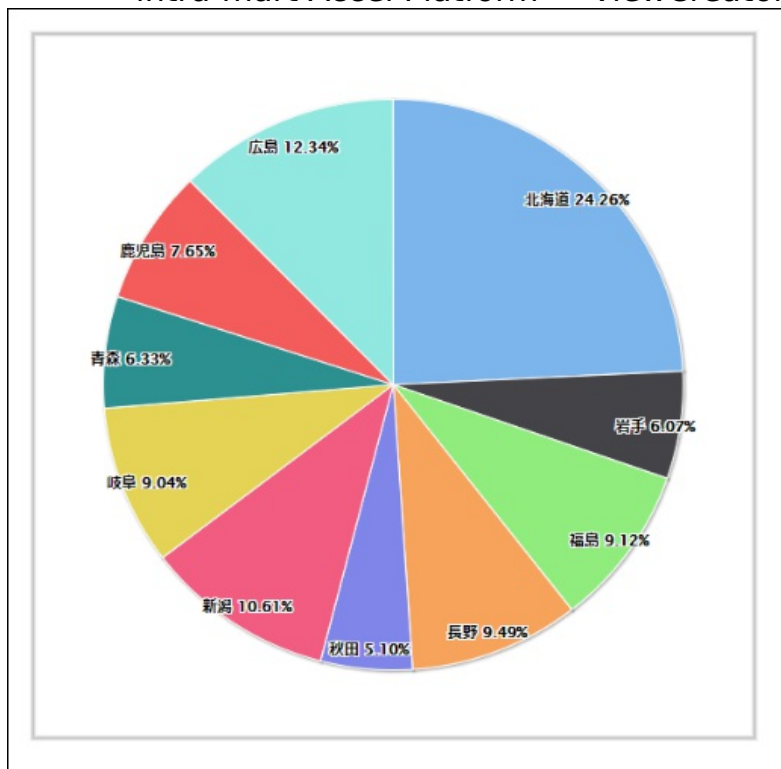


図 円グラフイメージ

エリアチャート

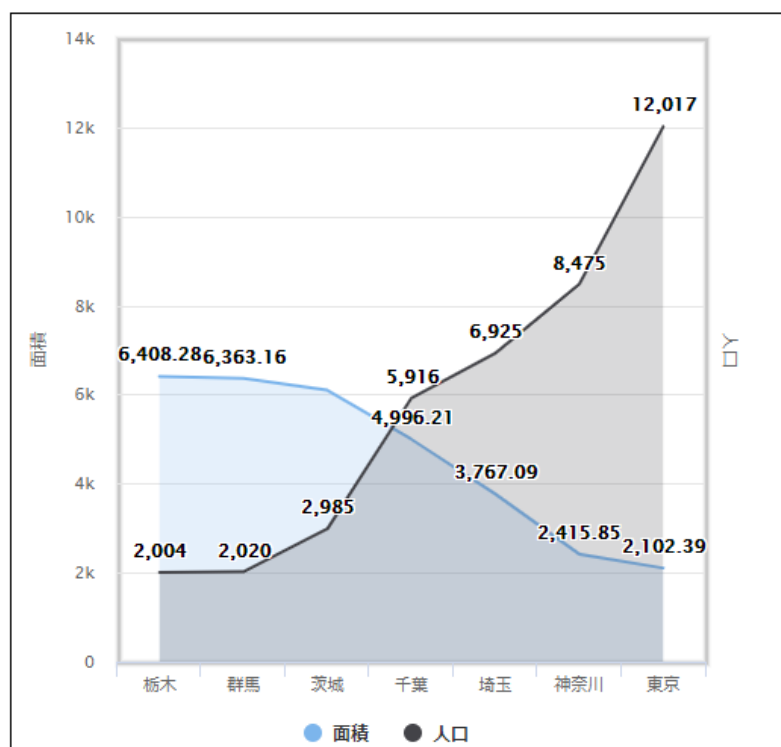


図 エリアチャートイメージ

グラフ設定

グラフ表示時のレイアウトを設定します。

グラフ設定	
グラフ描画形式*	Highcharts ▼
グラフの大きさ(縦幅)	500
グラフの大きさ(横幅)	500
100分率表示	<input checked="" type="checkbox"/>
リストの表示	<input checked="" type="checkbox"/>

図 データ参照・編集画面 グラフ設定項目

## 設定項目

項目名	説明
グラフ描画形式	グラフの描画形式を選択します。 intra-mart Accel Platform標準では「JFreeChart」、「Highcharts」2種類のグラフ描画形式を提供しています。
グラフの大きさ(縦幅)	グラフの表示領域(縦)を設定します。
グラフの大きさ(横幅)	グラフの表示領域(横)を設定します。
100分率表示	値カラムのデータの100分率をグラフ上に表示します。
リストの表示	グラフの数値データを表で添えるかどうかを設定します。



## 注意

グラフの大きさ(縦幅) / (横幅)の数値が小さい場合、グラフの表示領域が十分に見えてもデータの値が表示されないことがあります。

グラフの大きさ(縦幅) / (横幅)の数値を大きくすることで解消することがあります。

## カラム設定

グラフデータとして使用するデータが格納されているカラムを選択します。

クエリに追加されたカラムの内、数値型のカラムのみが表示されます。

データ毎に表示するグラフの種類を選択できます。

カラム設定						
カラムの国際化項目の編集						
キャプションカラム	地域名(name) ▼					
キャプション軸	<input checked="" type="radio"/> 横軸 <input type="radio"/> 縦軸					
凡例コードカラム	都道府県名(name) ▼					
凡例ラベルカラム	都道府県名(name) ▼					
表示	カラム	グラフタイプ	積み上げ表示	目盛り表示位置	カラー	
<input checked="" type="checkbox"/>	年度	棒グラフ ▼	グループ1 ▼	左側 ▼	#ffd4aa	
<input checked="" type="checkbox"/>	面積	折れ線グラフ ▼		右側 ▼	#aad4ff	
<input checked="" type="checkbox"/>	人口	円グラフ ▼				
<input type="checkbox"/>	人口密度	レーダーチャート ▼			#aafd4	

図 データ参照・編集画面 カラム設定項目

## 設定項目

項目名	説明
カラムの国際化項目の編集	カラムの国際化項目の編集ダイアログが表示されます。 詳細は、「 <a href="#">国際化とヘッダ・フッタ</a> 」を参照してください。
キャプションカラム	グラフのキャプションラベルとして使用するデータが格納されているカラムを選択します。
キャプション軸	キャプションカラムの表示軸を選択します。表示するグラフタイプに「 <a href="#">レーダーチャート</a> 」を選択している場合、キャプション軸の選択は無効です。
凡例コードカラム	詳細は、「 <a href="#">グラフ集計で「凡例」の表示方法を変更する</a> 」を参照してください。
凡例ラベルカラム	詳細は、「 <a href="#">グラフ集計で「凡例」の表示方法を変更する</a> 」を参照してください。
△▽アイコン	クリックで上下の順序を変更できます。
表示	選択したカラムがグラフ描画時に表示されます。
カラム	グラフを表示した場合の実データとして利用するカラムです。
グラフタイプ	データとして表示するカラムごとにグラフの種類を設定します。 Highchartsでは「 <a href="#">棒グラフ</a> 」、「 <a href="#">折れ線グラフ</a> 」、「 <a href="#">レーダーチャート</a> 」、「 <a href="#">円グラフ</a> 」、「 <a href="#">エリアチャート</a> 」5種類のグラフを提供しています。
積み上げ表示	同じグループに設定したカラムを積み上げて表示します。 グループは1から10までの振り分けを行えます。 グループの追加・削除は行えません。
目盛り表示位置	実データとして使用するカラムの目盛りの表示位置を設定します。 表示位置が同じカラムはグルーピングされます。
カラー	グラフ描画する際の色彩を設定します。

### コラム

- 円グラフ、折れ線グラフの場合、積み上げ表示は設定できません。
- 円グラフ、レーダーチャートの場合、目盛り表示位置は設定できません。
- 円グラフの場合、カラーは設定できません。
- 設定できない項目は非活性状態に変わります。

### 注意

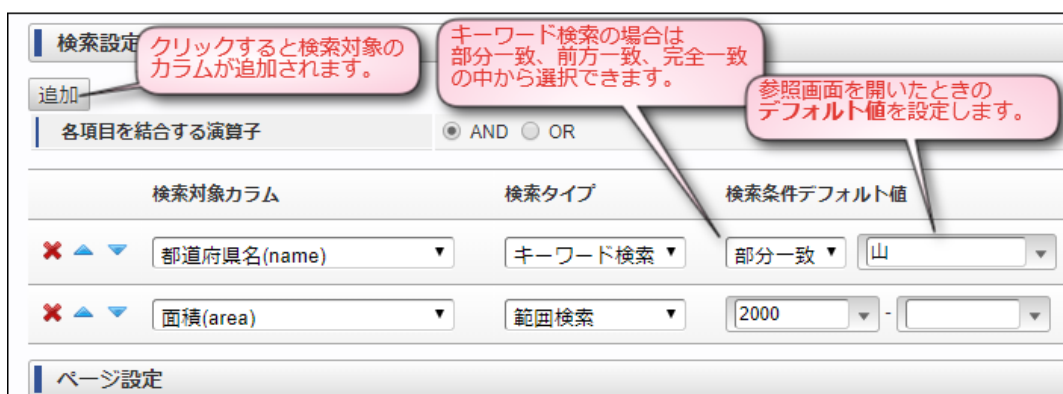
- 「Highcharts」は 2016 Summer(Nirvana)から利用可能です。
- グラフ描画形式の切り替えは 2016 Summer(Nirvana)から利用可能です。
- グラフの描画は利用するライブラリによって表示が異なる場合があります。

## 検索設定について

リスト集計、サマリ集計、クロス集計ではデータを参照画面で絞り込ませることができるよう設定できます。

### コラム

クロス集計の検索設定は2016 Spring(Maxima)から利用可能です。



The screenshot shows the search settings interface with the following elements and callouts:

- 検索設定** (Search Settings) section:
- 追加** (Add) button: Callout: "クリックすると検索対象のカラムが追加されます。" (Clicking adds the search target column.)
- 各項目を結合する演算子** (Operator): Radio buttons for AND and OR.
- 検索対象カラム** (Search Target Column): Dropdown menu with "都道府県名(name)" selected.
- 検索タイプ** (Search Type): Dropdown menu with "キーワード検索" (Keyword Search) selected.
- 検索条件デフォルト値** (Search Condition Default Value): Dropdown menu with "部分一致" (Partial Match) selected.
- 検索条件** (Search Condition): Input field with "山" (Yama) entered.
- 検索対象カラム** (Search Target Column): Dropdown menu with "面積(area)" selected.
- 検索タイプ** (Search Type): Dropdown menu with "範囲検索" (Range Search) selected.
- 検索条件** (Search Condition): Input field with "2000" entered.
- ページ設定** (Page Settings) section.

Additional callouts:

- Callout for "キーワード検索": "キーワード検索の場合は部分一致、前方一致、完全一致の中から選択できます。" (For keyword search, you can select partial match, front match, or full match.)
- Callout for "部分一致": "参照画面を開いたときのデフォルト値を設定します。" (Set the default value when opening the reference screen.)



上記のように設定しておく、参照画面を表示したときに「検索」がツールバーに表示されます。

■ リスト集計の場合

年度 - 地域 - 都道府県 グループ化リスト(人口)

← CSV出力 帳票出力 **検索**

各項目を結合する演算子  AND  OR

都道府県名 山  部分一致  前方一致  完全一致

面積 2000 -

検索 クリア

1

年度	地域名	都道府県名	年齢	人口(合計)
1995	中国	岡山	-	1,949
		山口	-	1,555
				3,504

■ クロス集計の場合

年度・都道府県別人口 - クロス集計

← 検索 帳票出力

入れ替え可能カラム一覧  
面積

列見出し  
年度 ソート指定無し  
年齢 ソート指定無し

行見出し  
地域名 見出し名 ソート指定無し  
都道府県名 見出し名 ソート指定無し

値  
人口 合計 数値フォーマット

表示/非表示  合計  平均  最小  最大  カウント

各項目を結合する演算子  AND  OR

地域名  すべて選択  
 北海道  東北  関東  中部  近畿  中国  四国  九州

人口 -

年度 1990 - 2000

表示 クリア

		1990			1995			2000			合計	平均
		0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上		
北海道	北海道	1,034	3,925	675	899	3,943	845	792	3,833	1,032	16,978	1,886.44
	小計										16,978	1,886.44
東北	青森	289	1,001	192	252	991	237	223	965	287	4,437	493
	岩手	270	941	206	239	925	255	212	899	304	4,251	472.33
	宮城	439	1,535	267	394	1,596	338	354	1,602	409	6,934	770.44
	秋田	220	816	192	189	787	238	163	746	280	3,631	403.44
	山形	234	819	205	209	799	249	186	772	286	3,759	417.67
	福島	422	1,378	302	382	1,380	372	341	1,354	432	6,363	707
	小計										29,375	543.98

**i** コラム

画面上で「検索（表示）」を実施していない場合でも、「CSV出力」「帳票出力」クリック時に検索設定に入力がある場合は、検索設定の絞り込み結果が出力されます。

**i** コラム

日付型、タイムスタンプ型の検索設定の条件値に、日付に変換できない文字列が入力された場合は、システム日付で検索が実行されます。

## AND検索とOR検索（各項目を結合する演算子）について

検索項目が複数の場合は、AND検索かOR検索を選択できます。

- AND検索の場合

検索項目の条件をすべて満たすレコードが検索結果として表示されます。  
ただし、条件値が入力されていない検索項目を除きます。

- OR検索の場合

検索項目のうち、いずれかを満たすレコードが検索結果として表示されます。  
ただし、条件値が入力されていない検索項目を除きます。

**i** コラム

「範囲検索」を設定した場合、検索式は「 下限値 ≤ 設定値 ≤ 上限値」で検索されます。

**i** コラム

クロス集計では、常に「検索」がツールバーに表示されます。

**i** コラム

検索設定の初期値として、動的に生成された値をセットすることも可能です。  
具体的にセット可能な値については、「[動的パラメータ](#)」を参照してください。

The screenshot shows a search interface with a dropdown menu open. The dropdown menu contains the following options:

- <%ENCODING%>
- <%LOCALE%>
- <%SYSTEM\_DATE%>
- <%SYSTEM\_DATE/yyyy/MM/dd HH:mm%>
- <%SYSTEM\_DATE/yyyy/MM/dd%>
- <%SYSTEM\_DATE\_TIMEZONE/yyyy/MM/dd HH:mm%>
- <%USER\_ID%>

The interface also shows a search type dropdown set to 'キーワード検索', a search mode dropdown set to '部分一致', and a search value dropdown set to '山'. A button labeled '登録して一覧へ戻る' is visible at the bottom.

**i** コラム

2019 Spring(Violette)より、最後に表示したデータの検索条件やソート順などの表示情報が保存されます。  
保存した情報は「ブラウザを閉じる」、「リロードアイコンをクリックする」、「データ参照の変更が行われる」のいずれかの操作によって破棄されます。

**i** コラム

OR検索は2019 Spring(Violette)より、利用できます。

## リスト集計とサマリ集計で利用可能な機能

検索の条件値入力に選択型コンポーネントを利用する

検索条件値はテキストフィールドでの入力だけでなく、「セレクトボックス」、「コンボボックス」、「オートコンプリート」を利用できます。

**i** コラム

コンボボックスの利用は2014 Summer(Honoka)より可能です。  
 セレクトボックスとオートコンプリートの利用は2018 Summer(Tiffany)より可能です。

**ユーザー一覧サンプル**

← CSV出力 帳票出力 検索

各項目を結合する演算子  AND  OR

user\_cd

locale\_id   部分一致  前方一致  完全一致

user\_cd   部分一致  前方一致  完全一致

検索 クリア

1 2 ▶▶▶

user_cd	locale_id	start_date	user_name
tenant	zh_CN	1900/01/01	tenant
tenant	en	1900/01/01	tenant
tenant	ja	1900/01/01	tenant
aoyagi	ja	1900/01/01	青柳辰巳
aoyagi	ja	2000/01/01	青柳辰巳
hayashi	ja	1900/01/01	林政義
hayashi	ja	2000/01/01	林政義
maruyama	ja	1900/01/01	円山益男
maruyama	ja	2000/01/01	円山益男
sekine	ja	1900/01/01	関根千香

オートコンプリートを利用した場合の例。

**ユーザー一覧サンプル**

← CSV出力 帳票出力 検索

各項目を結合する演算子  AND  OR

user\_cd

locale\_id   部分一致  前方一致  完全一致

user\_cd   部分一致  前方一致  完全一致

検索 クリア

1 2 ▶▶▶

user_cd	locale_id	start_date	user_name
tenant	zh_CN	1900/01/01	tenant
tenant	en	1900/01/01	tenant

■ 設定方法

「検索設定」で「検索条件値・選択肢リスト」を「有効」にします。

The screenshot shows the '検索設定' (Search Settings) dialog. At the top, there is a '追加' (Add) button and a section for '各項目を結合する演算子' (Operator for combining items) with radio buttons for 'AND' and 'OR'. Below this is a table with columns: '検索対象カラム' (Search Target Column), '検索タイプ' (Search Type), '検索条件デフォルト値' (Search Condition Default Value), and '検索条件値・選択肢リスト' (Search Condition Value / Selection List). Two rows are visible: one for 'user\_cd(user\_cd)' and one for 'locale\_id(locale\_id)'. Both rows have 'キーワード検索' (Keyword Search) as the search type and '部分一致' (Partial Match) as the default condition. The '有効' (Active) toggle switch for each row is highlighted with a red box.

「選択肢の設定」ボタンを押下すると、ダイアログが表示されます。

The screenshot shows the '選択肢の設定 - user\_cd' (Selection Item Settings - user\_cd) dialog box. It has a title bar with a close button. The dialog is divided into several sections: '設定方法' (Setting Method) with radio buttons for '固定値を入力' (Enter fixed value), 'データ参照を利用' (Use data reference), and '計算式を利用' (Use formula); 'データ参照' (Data Reference) with a text input field 'データ参照名で検索します' (Search by data reference name) and a checked checkbox '編集中のデータ参照を利用する' (Use data reference in editing); '表示値' (Display Value) with a dropdown menu set to 'user\_name'; '送信値' (Send Value) with a dropdown menu set to 'user\_cd'; 'データの扱い' (Data Handling) with checked checkboxes for '重複を除外' (Exclude duplicates) and '空白を除外' (Exclude blanks); and 'コンポーネントの種類' (Component Type) with radio buttons for 'セレクトボックス' (Select box), 'コンボボックス' (Combo box), and 'オートコンプリート' (Auto-complete). An 'OK' button is at the bottom right.

■ 共通の設定

設定方法

選択肢として表示する一覧データの取得方法を選択します。

【固定値を入力】

選択肢として表示する項目を手入力で設定します。

【データ参照を利用】

選択肢として表示する項目を、リスト集計または、サマリ集計のデータ参照を利用して生成します。編集中のデータ参照だけでなく、他のデータ参照を利用することも可能です。指定されたデータ参照に対して参照権が無いユーザが表示した場合、選択肢の生成が行われません。

【計算式を利用】

選択肢として表示する項目を計算式で生成します。

データの扱い

【重複を除外】

「送信値」が重複する選択肢を除外します。

【空白を除外】

「送信値」が null または空文字の選択肢を除外します。

コンポーネントの種類

【セレクトボックス】

セレクトボックスで表示します。

【コンボボックス】

コンボボックスで表示します。項目を選択すると、「送信値」が表示されます。

【オートコンプリート】

テキストフィールドで表示します。「表示値」に対して入力された文字列の部分一致で絞り込まれた一覧を生成します。項目の選択が行われなかった場合は、テキストフィールドに入力された文字列がそのまま検索条件値として利用されます。

■ 「データ参照を利用」が選択された場合の設定

データ参照

選択肢の生成に利用するデータ参照を設定します。リスト集計または、サマリ集計のデータ参照を設定します。

表示値

選択肢の表示値（ラベル）として利用するデータを取得するカラムを選択します。

送信値

選択肢の送信値として利用するデータを取得するカラムを選択します。

■ 「固定値を入力」が選択された場合の設定

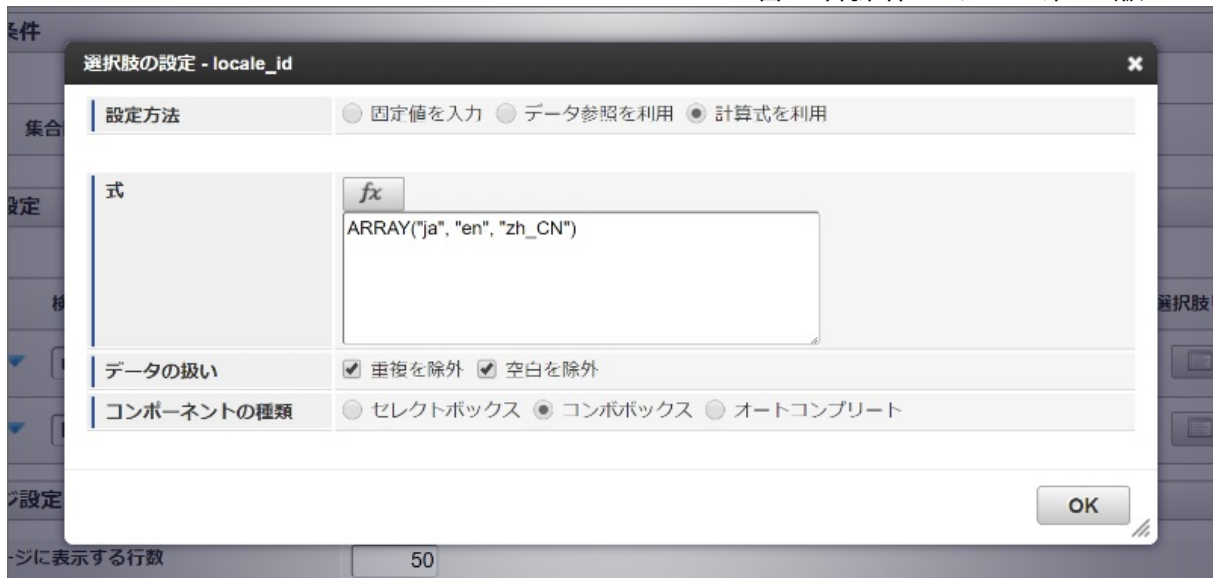
「表示値」と「送信値」を手入力で設定します。



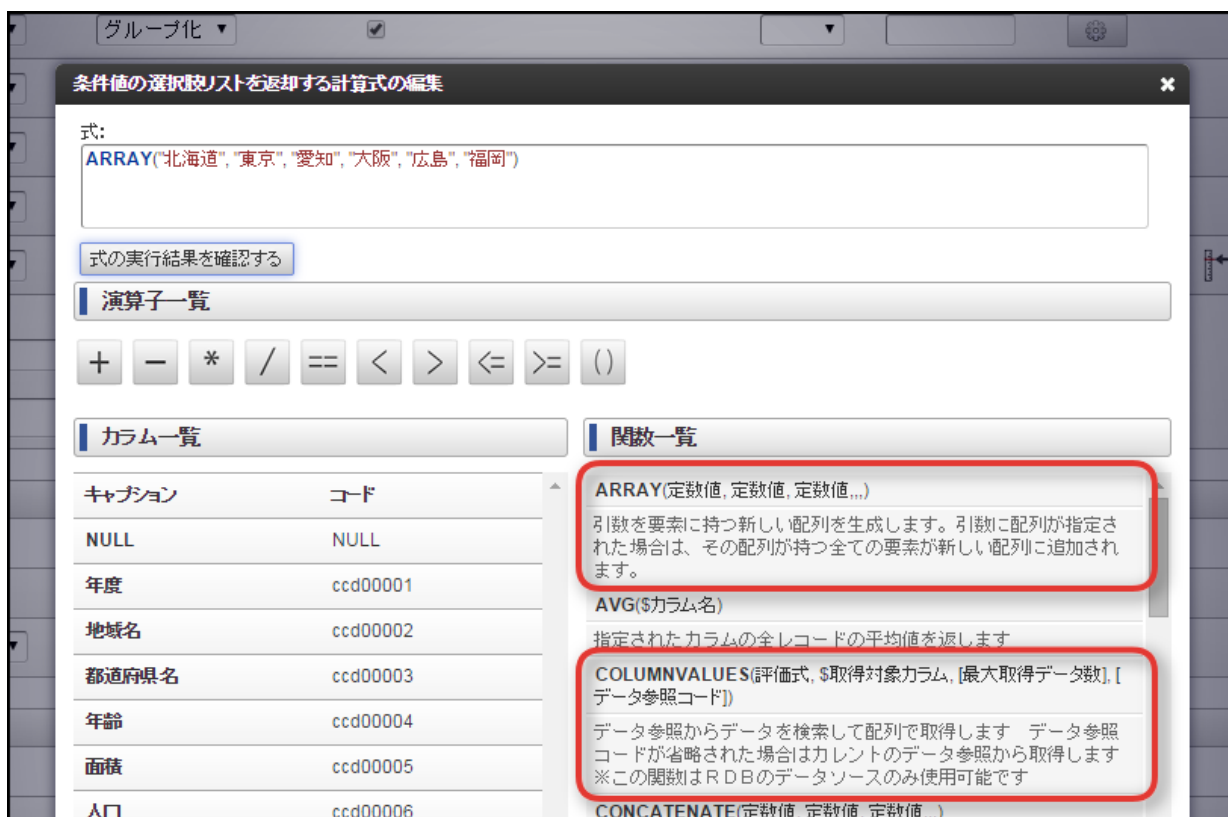
■ 「計算式を利用」が選択された場合の設定

「式」に選択肢として利用するデータを返却する計算式を設定します。

「表示値」と「送信値」を分けて設定することはできません。



関数ボタンをクリックすると、計算式入力用のダイアログが表示されます。



計算式の結果が「配列型」となるように設定してください。

配列型を返却する関数として、ARRAY関数やCOLUMNVALUES関数が用意されています。

コンボボックスに表示する選択肢を動的に生成する場合は、COLUMNVALUES関数を利用してください。

### コラム

計算式や関数についての詳細は、「[利用可能な関数一覧](#)」を参照してください。

## クロス集計で利用可能な機能

リスト選択（チェックボックスによる絞り込み）

クロス集計の検索設定では、チェックボックス形式で表示する値を選択させることが可能です。

年度・都道府県別人口 - クロス集計

検索 帳票出力

入れ替え可能カラム一覧

面積

列見出し

年度 ソート指定無し

年齢 ソート指定無し

行見出し

地域名 見出し名 ソート指定無し

都道府県名 見出し名 ソート指定無し

値

人口 合計 数値フォーマット

表示/非表示  合計  平均  最小  最大  カウント

各項目を結合する演算子  AND  OR

地域名  すべて選択

北海道  東北  関東  中部  近畿  中国  四国  九州

人口  -

年度  1990 -  2000

表示 クリア

		1990			1995			2000			合計	平均
		0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上		
北海道	北海道	1,034	3,925	675	899	3,943	845	792	3,833	1,032	16,978	1,886.44
	小計										16,978	1,886.44
東北	青森	289	1,001	192	252	991	237	223	965	287	4,437	493
	岩手	270	941	206	239	925	255	212	899	304	4,251	472.33
	宮城	439	1,535	267	394	1,596	338	354	1,602	409	6,934	770.44
	秋田	220	816	192	189	787	238	163	746	280	3,631	403.44
	山形	234	819	205	209	799	249	186	772	286	3,759	417.67
	福島	422	1,378	302	382	1,380	372	341	1,354	432	6,363	707
	小計											29,375

■ 設定方法

データ参照編集画面で、「検索タイプ」を「リスト選択」に設定してください。

**検索設定**

追加

各項目を結合する演算子  AND  OR

検索対象カラム	検索タイプ	検索条件デフォルト値
<input checked="" type="checkbox"/> ▲ ▼ 地域名(name)	<input type="text" value="リスト選択"/>	
<input checked="" type="checkbox"/> ▲ ▼ 人口(population)	<input type="text" value="キーワード検索"/>	<input type="text" value="部分一致"/>

検索対象カラムに含まれるデータについて、重複を取り除いた形でチェックボックスが生成されます。



**注意**

日付型、タイムスタンプ型、バイナリ型、真偽値型のカラムは「リスト選択」では利用できません。

**参照権の設定**

参照権の追加ボタンで、このデータ参照でデータの閲覧を許可するユーザ、ロール、会社/組織、グループを設定します。

ゲストユーザ、認証済みユーザに対して参照を許可する場合はチェックボックスを選択します。



**コラム**

参照権の設定は2014 Spring(Granada)から利用可能です。

1ページに表示する行数

ヘッダー

fx

設定された行数(レコード数)ごとにページングします。

U S 書式 段落 フォントファミリー フォントサイズ

国際化データ

フッター

fx

B I U S 書式 段落 フォントファミリー フォントサイズ

国際化データ

権限設定

ゴミ箱アイコン

参照権 追加

- 青柳辰巳
- 認可管理者
- サンプル課11
- サンプル課21
- パブリックグループa
- パブリックグループc

ゲストユーザ  認証済みユーザ

登録して一覧へ戻る

**i** コラム

参照権を設定しなかった場合、そのデータ参照の作成者および、管理者のみが閲覧できます。

**i** コラム

参照権はゴミ箱アイコンをクリックすることで削除できます。

設定が終わったら登録ボタンをクリックします。



フッター

fx

書式 段落 フォントファミリー フォントサイズ

国際化データ

権限設定

参照権 追加

青柳辰巳
   
 認可管理者
   
 サンプル課11
   
 サンプル課21
   
 パブリックグループa
   
 パブリックグループc

登録ボタン

登録して一覧へ戻る

ゲストユーザ  認証済みユーザ

### ! 注意

既存のデータ参照の編集は、管理者ユーザかそのクエリを作成したユーザのみが実行できます。

### i コラム

データ参照を閲覧するためには、認可の設定も必要です。

必要に応じて設定を追加、変更してください。

詳細は「[ViewCreator全体の権限設定](#)」を参照してください。

ここでは詳細機能についてご紹介します。

## 抽出条件の設定

「クエリ編集」画面、およびリスト集計の「データ参照編集」画面では、レコードの抽出条件を設定できます。

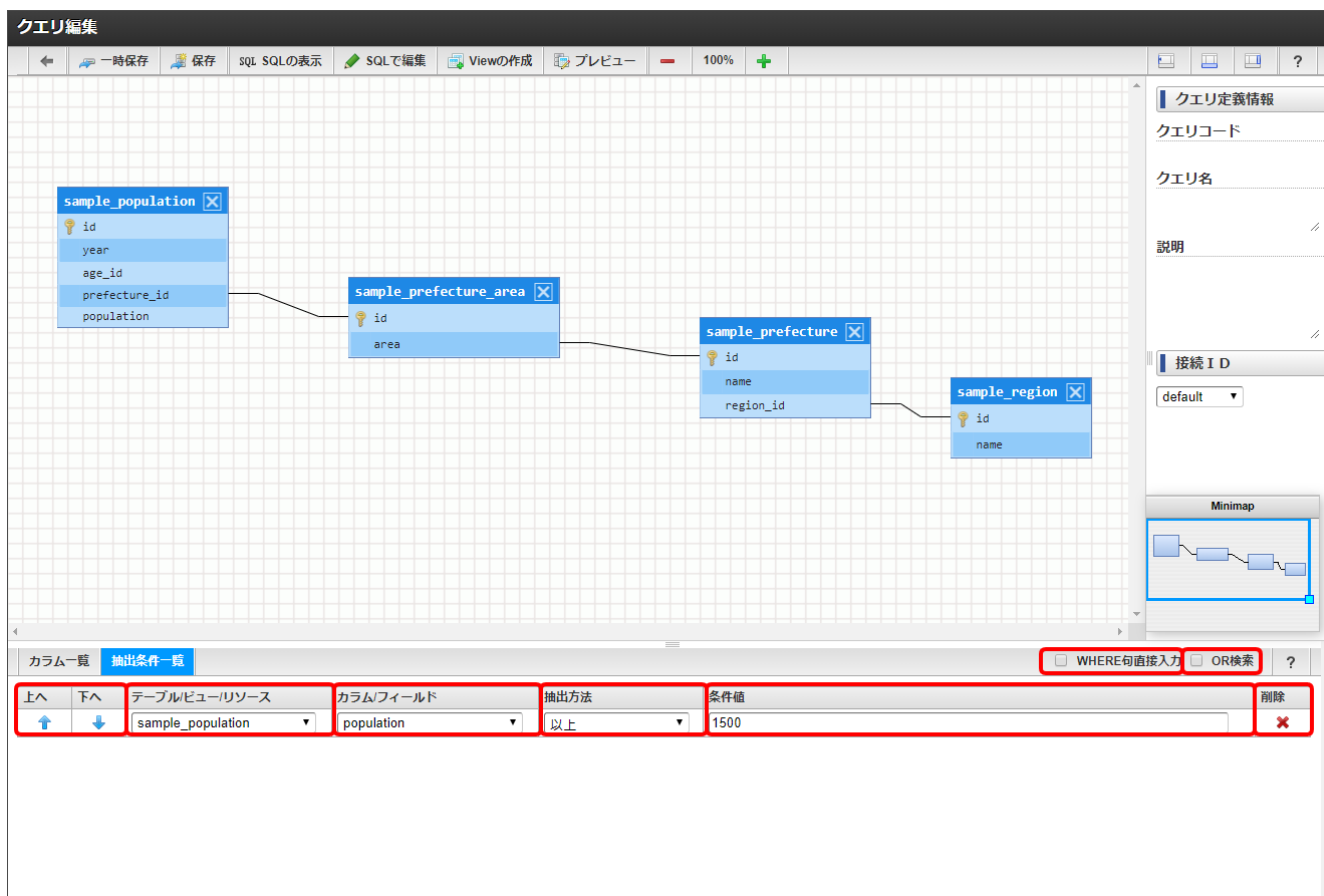
リスト集計の「検索設定」は、データ参照を閲覧するユーザが目的の情報に素早く辿り着くための機能ですが、それに対して「抽出条件」は、ユーザに見せたくない情報を設定者側であらかじめ絞っておくための機能です。

### 目次

- クエリ作成時に抽出条件を設定する
  - 抽出条件を直接入力する
- データ参照作成時に抽出条件を設定する
- データ参照時のSQL
- 動的パラメータ

## クエリ作成時に抽出条件を設定する

「クエリ編集」画面の「抽出条件一覧」タブからレコードの抽出条件を設定します。



図：クエリ編集画面 - 抽出条件

項目名	説明
WHERE句直接入力	抽出条件を設定する代わりに適用するWHERE句を入力します。 動的パラメータを埋め込むことも可能です。

項目名	説明
OR検索	複数の抽出条件が設定された場合に OR 条件でデータ抽出を行います。 チェックが外されている場合は AND 条件が適用されます。
上へ下へ	抽出条件の順序を入れ替えます。
テーブル/ビュー/リソース	デザインに追加されているエンティティを選択可能です。
カラム/フィールド	抽出条件の対象とするカラムを選択します。 「テーブル/ビュー/リソース」で選択されたエンティティが持つカラムを選択可能です。
抽出方法	部分一致、範囲指定などの抽出方法を指定します。
条件値	固定の条件値だけでなく、カレントのユーザコードなどの動的に変化する値を指定することも可能です。
削除	抽出条件一覧からカラムを削除します。

### 抽出条件を直接入力する

複雑な条件式を設定したい場合は抽出方法を「直接入力」にします。

直接入力を選択されると、算術演算子の挿入が行われないため演算子を適宜入力してください。

抽出方法	条件値
直接入力 完全一致 部分一致 前方一致 以上 以下 未滿 範囲指定 直接入力	IN('長野','岩手','福島','大分')

図：クエリ編集画面 - 直接入力

WHERE句直接入力にチェックを入れることで、WHERE句そのものを直接記述することも可能です。

カラム一覧	抽出条件一覧	<input checked="" type="checkbox"/> WHERE句直接入力	<input type="checkbox"/> OR検索
where area < 5000			

図：クエリ編集画面 - WHERE句直接入力



#### 注意

直接入力を使用する場合は、テーブルやレコードを壊すことの無いように細心の注意を払ってください。

### データ参照作成時に抽出条件を設定する

リスト集計のデータ参照作成時に、「データ参照編集」画面からレコードの抽出条件を設定します。

クロス集計とグラフ集計では設定できません。



図：データ参照作成画面 - 抽出条件

項目名	説明
追加	抽出条件を追加します。
「削除」アイコン	抽出条件一覧からカラムを削除します。
「上へ下へ」アイコン	抽出条件の順序を入れ替えます。
集合関数	抽出パターンを指定します。 リスト集計で設定可能なのは「1レコード単位」のみです。
対象カラム	抽出条件の対象とするカラムを選択します。 選択可能なカラムが表示されます。
抽出方法	部分一致、範囲指定などの抽出方法を指定します。
条件値	固定の条件値だけでなく、カレントのユーザコードなどの動的に変化する値を指定することも可能です。

### コラム

「1レコード単位」以外の抽出パターンは、サマリ集計を選択している場合のみ設定可能です。

SQL上では、下記のように適用されます。

1レコード単位 : WHERE句

集合関数 : HAVING句

### 注意

リスト集計の「抽出条件」では、OR検索は設定できません。

## データ参照時のSQL

データ参照が閲覧されたときに発行されるSQLは次のような構造です。

SELECT

[データ参照編集画面で表示されているカラム一覧]

FROM

( SELECT

[クエリに追加されたカラム一覧]

FROM

[クエリに追加されたテーブル]

WHERE

[クエリで設定された抽出条件]

) IM\_QUERY\_\_

WHERE

[データ参照で設定された抽出条件]

データベースの実装に依存しますが、一般的にはクエリの段階でデータを絞っておいた方がSQLの実行速度は良くなる傾向にあります。ただし、1つのクエリを共有して複数のデータ参照を作成し、抽出条件はデータ参照ごとに設定した方が良いケースもあるので、適宜使い分けて設定してください。

## 動的パラメータ

抽出条件の値には固定値だけでなく動的に変化する値などの特別な値をセットすることも可能です。

条件値
<%ENCODING%>
<%LOCALE%>
<%NOT_NULL%>
<%NULL%>
<%REQUEST_PARAMETER%>
<%REQUEST_PARAMETER(KEY_NAME)%>
<%SYSTEM_DATE%>
<%SYSTEM_DATE(yyyy/MM/dd HH:mm)%>
<%SYSTEM_DATE(yyyy/MM/dd)%>
<%SYSTEM_DATE_TIMEZONE(yyyy/MM/dd HH:mm)%>
<%USER_ID%>

図：クエリ編集画面 - 抽出条件

図：データ参照編集画面 - 抽出条件

パラメータ名	生成される値
<%DYNAMIC_PARAMETER%>	API経由でパラメータ値をセットする場合に使用します。  このパラメータが設定された抽出条件が1つでも存在するクエリからは、データ参照を作成できません。  詳しくは、 <a href="#">jp.co.intra_mart.foundation.viewcreator.model.Query#execute</a> を参照してください。
<%ENCODING%>	SQL実行時のログインユーザに関連づけられたエンコーディング文字列がセットされます。
<%LOCALE%>	SQL実行時のログインユーザに関連づけられたロケール文字列がセットされます。
<%NOT_NULL%>	IS NOT NULL がセットされます。  ※このパラメータを選択した場合は、必ず「抽出方法」で「完全一致」を選択してください。
<%NULL%>	IS NULL がセットされます。  ※このパラメータを選択した場合は、必ず「抽出方法」で「完全一致」を選択してください。
<%SYSTEM_DATE(*)%>	SQL実行時のシステム日付がセットされます。  システム日付データを指定されたフォーマットに従った、文字列がセットされます。  文字列で生成された日付データは、データベースごとに用意された型変換関数に渡されます。  ※このパラメータは、必ず日付型またはタイムスタンプ型のカラムに対してセットしてください。
<%USER_ID%>	SQL実行時のログインユーザコードがセットされます。
<%REQUEST_PARAMETER(KEY_NAME)%>	HttpServletRequest (リクエスト)、または、RenderRequest (ポートレットのリクエスト) から取得した値がセットされます。 取得される優先順位は、HttpServletRequestが優先されて取得されます。  「KEY_NAME」にはパラメータ名を指定します。

**i** コラム

<%REQUEST\_PARAMETER(KEY\_NAME)%>は2014 Spring(Granada)から利用可能です。  
2016 Spring(Maxima)から、<%REQUEST\_PARAMETER(KEY\_NAME)%>でRenderRequest (ポートレットのリクエスト) が取得可能になりました。

**i** コラム

- <%REQUEST\_PARAMETER(KEY\_NAME)%>の使用例

1. 検索設定の場合

タグが受け取る引数として、リクエストパラメータのキー名を任意に指定します。

検索設定

追加

各項目を結合する演算子  AND  OR

検索対象カラム	検索タイプ	検索条件デフォルト値	検索条件値・選択肢リスト
都道府県名(name)	キーワード検索	部分一致	<%REQUEST_PAI
面積(area)	範囲検索		

データ参照を表示した際に指定したリクエストパラメータを受け取ることができます。

```
http://.../viewcreator/dataview/list/{dataview_cd}?param=島
```

2000年度版・地域 - 都道府県 サマリ集計

CSV出力 帳票出力 検索

各項目を結合する演算子  AND  OR

都道府県名   部分一致  前方一致  完全一致

面積  -

検索 クリア

地域名	都道府県名	面積(平均)	人口(合計)
四国	徳島	4,145	824
中国	島根	6,708	761
中国	広島	8,478	2,877
九州	鹿児島	9,133	1,785
東北	福島	13,783	2,127
		8,449	8,374

2. 抽出条件の場合

タグが受け取る引数として、リクエストパラメータのキー名を任意に指定します。

抽出条件

追加

集合関数	対象カラム	抽出方法	条件値
1レコード単位	地域名(name)	部分一致	<%REQUEST_PARAMETER
1レコード単位	都道府県名(name)	部分一致	<%REQUEST_PARAMETER

データ参照を表示した際に指定したリクエストパラメータを受け取ることができます。

http://.../viewcreator/dataview/list  
/{dataview\_cd}?region=関東&pref=東京

2000年度版・地域 - 都道府県 サマリ集計

CSV出力 帳票出力 検索

地域名	都道府県名	面積(平均)	人口(合計)	
関東	東京	2,102	11,409	★
関東	東京	2,102	11,656	★
関東	東京	2,102	11,598	★
関東	東京	2,102	11,819	★
関東	東京	2,102	11,762	★
関東	東京	2,102	11,735	★
関東	東京	2,102	12,017	★
		2,102	81,996	

3. ポートレットで利用する場合  
メニューから「サイトマップ」→「ポータル - ポートレット一覧」を選択します。  
編集アイコンをクリックし、「ポートレット編集」画面を開きます。  
基本設定のページ引数に抽出条件で設定したタグのキー名を指定します。

基本設定

ポートレット名 スクリプト開発モデルポートレット

名称\*

日本語*	2000年度版・地域 - 都道府県 サマリ集計
英語*	2000-Regions-Prefectures summary
中国語(中華人民共和国)*	2000年度版本 区域 - 都道府県 统计

アプリケーション

日本語	ViewCreator
英語	ViewCreator
中国語(中華人民共和国)	ViewCreator

ページパス\* viewcreator/views/dataview/list4portal

ページ引数 portlet=true&dataview\_cd=5i7urzmesvqqip4&region=関東

Actionハンドラ

Eventハンドラ

「ポートレット編集」画面のページ引数で指定したリクエストパラメータを受け取ることができます。

2000年度版・地域 - 都道府県 サマリ集計

詳細

年度	地域名	都道府県名	年齢	面積(平均)	人口(合計)	
1975	関東	東京	○	2,102	11,656	★
1995	関東	東京	○	2,102	11,735	★
1980	関東	東京	○	2,102	11,598	★
2000	関東	東京	○	2,102	12,017	★
1985	関東	東京	○	2,102	11,819	★
1970	関東	東京	○	2,102	11,409	★
1990	関東	東京	○	2,102	11,762	★
1970	関東	神奈川県	○	2,416	5,473	★
1995	関東	神奈川県	○	2,416	8,238	★
1980	関東	神奈川県	○	2,416	6,919	★
1975	関東	神奈川県	○	2,416	6,394	★
1990	関東	神奈川県	○	2,416	7,955	★
1985	関東	神奈川県	○	2,416	7,429	★
2000	関東	神奈川県	○	2,416	8,475	★
1980	関東	埼玉県	○	3,767	5,417	★
1975	関東	埼玉県	○	3,767	4,819	★
1995	関東	埼玉県	○	3,767	6,748	★
1970	関東	埼玉県	○	3,767	3,867	★



**i** コラム

- <%SYSTEM\_DATE%>で、1時間前、前日などの指定をする

パラメータ<%SYSTEM\_DATE(yyyy/MM/dd HH:mm)%>は、実行時のシステム日付がセットされますがフォーマット文字列部分に、+演算子や-演算子を用いた記述が可能です。

- システム日付の1時間前をセットしたい場合

```
<%SYSTEM_DATE(yyyy/MM/dd HH-1:mm)%>
```

- システム日付の前日をセットしたい場合

```
<%SYSTEM_DATE(yyyy/MM/dd-1 HH:mm)%>
```

※演算子には、+と-のみ使用できます。

※演算子の前後にスペースは入れないようにしてください。

**i** コラム

下記のファイルを編集することでオリジナルの動的パラメータを作成することもできます

WEB-INF/conf/viewcreator-config.xml

例) LOCALEパラメータの設定と実装例

```
<param>
  <param-name>LOCALE</param-name>
  <class-name>jp.co.intra_mart.foundation.viewcreator.params.SystemParameterLocale</class-name>
</param>
```

パラメータ名 (<param-name>タグの値) には半角大文字の英字とアンダースコアのみを使用します。

<class-name>の値にはjp.co.intra\_mart.foundation.viewcreator.SystemParameterインタフェースの実装クラスを指定します。

```
public class SystemParameterLocale implements SystemParameter {

  @Override
  public String get(final String syskey) {
    final AccountContext context = Contexts.get(AccountContext.class);
    return context.getLocale().toString();
  }
}
```

上記変更を反映するにはサーバの再起動が必要です。

## 外部ページへの連携 (リスト/サマリ集計のみ)

「リスト集計」では、レコードを他ページへのリンク表示にできます。

年度	都道府県名	地域名	面積	人口
1970	茨城	関東	6096	534
1970	栃木	関東	6408	389
1970	群馬	関東	6363	397
1970	埼玉	関東	3767	1003
1970	千葉	関東	4996	846
1970	東京	関東	2102	2401
1970	神奈川	関東	2416	1302
1975	茨城	関東	6096	580

こうすることで、ViewCreatorをオリジナルアプリケーションの一覧画面として使用できます。

#### 目次

- キー名の変更
- データ型と送信データについて
- PathVariablesへの対応

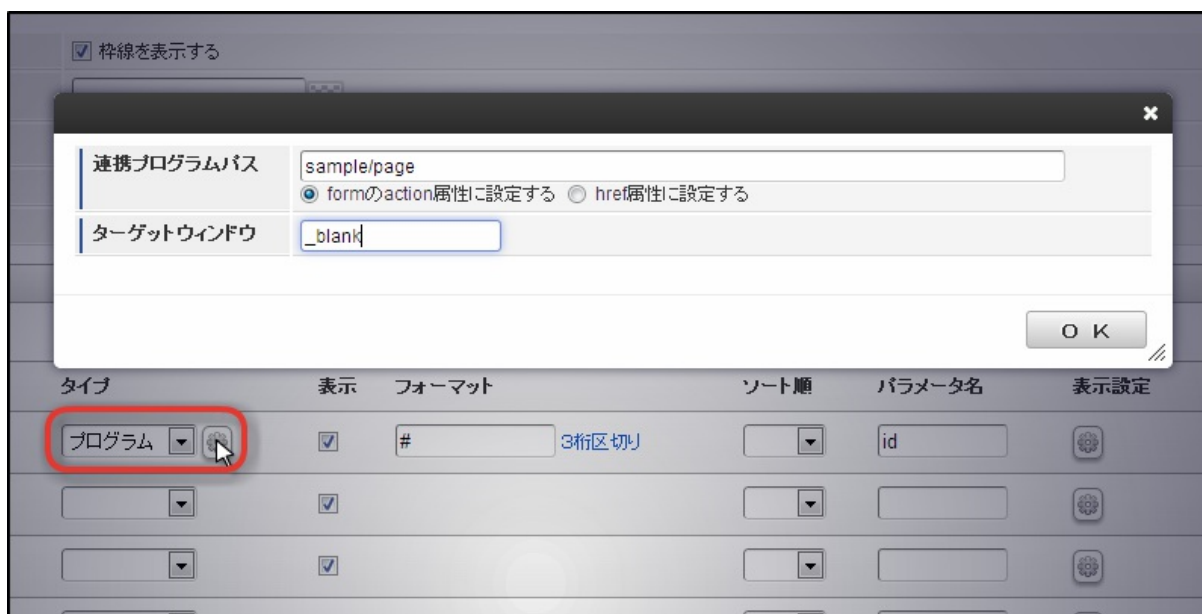
- 設定方法

データ参照編集画面を開き、次の2つの項目を設定します。

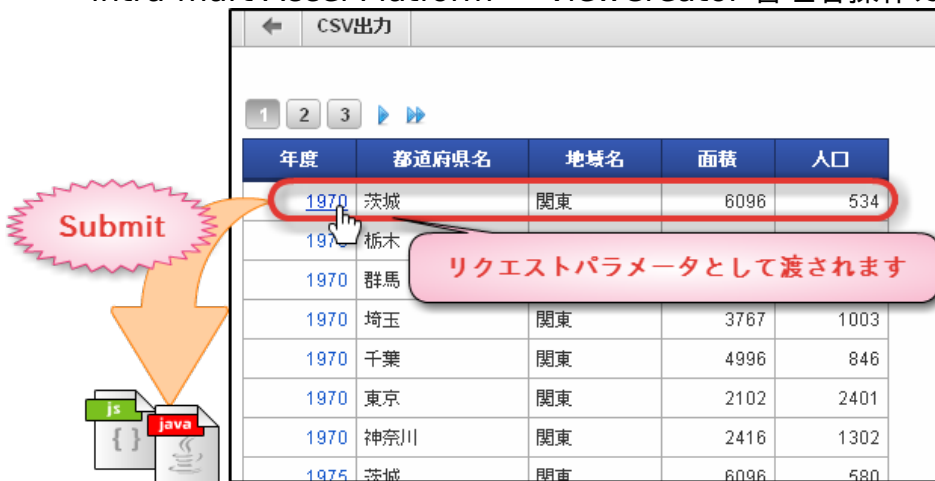
#### # 連携プログラムパス # ターゲットウィンドウ

- リンク表示させるカラムの「タイプ」を「プログラム」に設定します。

タイプを「プログラム」に設定すると、プログラムパス入力用ダイアログを表示するボタンが表示されます。



リンクがクリックされると、連携プログラムパスで設定されたプログラムへ連携されます。



クリックされたレコードのデータがリクエストパラメータとして連携先のプログラムへ渡されます。

リクエストパラメータのキーはデータベース上のフィールド名が設定されます。

### キー名の変更

キー名は任意の値に変更できます。



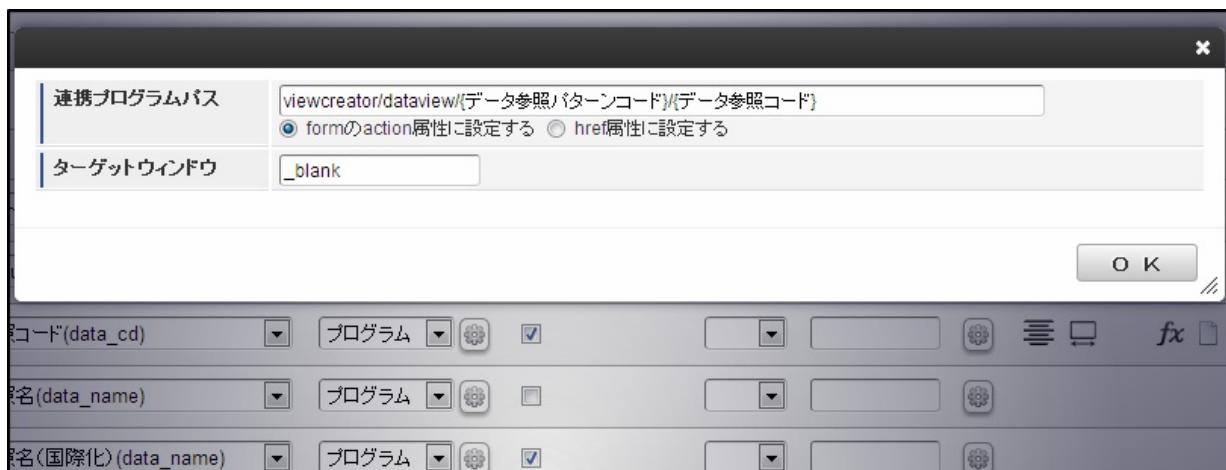
### データ型と送信データについて

データ型	説明
文字列型	カラムに格納されているデータをそのまま送信します。
数値型	カラムに格納されているデータの文字列表現を送信します。 フォーマット文字列の適用は行いません。
日付型	ミリ秒に変換された数値文字列データと、フォーマット文字列を適用したデータを送信します。 上記のデータは別々のパラメータとして送信され、後者のパラメータのキー名には「_format」が追加されます。 フォーマット文字列はデータ参照の定義に含まれるカラムの「フォーマット」の設定値が適用されます。 フォーマット文字列が未設定または空文字の場合は、JavaScriptのDateオブジェクトの文字列表現が送信されます。
タイムスタンプ型	日付型と同じです。
バイナリ型	送信しません。
真偽値型	「true」または「false」を送信します。

#### コラム

null 値の場合はデータ型に関係なく、空文字が送信されます。  
※ただし、バイナリ型を除きます。

パスに含まれる文字列は、固定値を設定するだけでなく、レコードから取得した値を埋め込むことも可能です。



#### ■ フォーマット

カラムを表す文字列（キャプション名、またはカラムコード）を、{ と } で括ることで実行時に値が展開されます。

{%キャプション名%} または {%カラムコード%}

設定例	実行時に設定される値の例
viewcreator/dataview/{データ参照パターンコード}/{データ参照コード}	viewcreator/dataview/list/dataview_list_sample

#### コラム

PathVariablesについての詳細は「[スクリプト開発モデル プログラミングガイド - PathVariables](#)」を参照してください。

## ポータルとの連携

作成したデータ参照はポートレットとして登録することで、ポータル画面へ表示できます。ここでは、その手順についてご紹介します。

#### コラム

2016 Spring(Maxima)から、ポートレットに対して<%REQUEST\_PARAMETER(KEY\_NAME)%>が利用可能になりました。詳細は「[動的パラメータ](#)」を参照してください。

ポータルに表示したいデータ参照を確認して「ポートレット登録」を選択します。

データ参照一覧

コピー 削除 ソースコード出力 CSV出力ジョブの登録 **ポータル登録** クエリー一覧へ

検索 クリア

編集	データ参照	データ参照名	更新日	作成者
<input type="checkbox"/>		2000年度版・地域 - 都道府県 サマリ	2013/11/05 22:18	tenant
<input type="checkbox"/>		2000年度版・地域 - 都道府県 サマリ	2013/10/31 18:20	tenant
<input type="checkbox"/>		[ポータル用]トランジションログ・ページ	2014/06/13 01:16	tenant
<input type="checkbox"/>		エラーログ	2014/06/13 01:16	tenant
<input type="checkbox"/>		サマリ集計から作成した棒グラフサン	2013/11/06 16:29	tenant
<input type="checkbox"/>		システムログ	2014/06/13 01:16	tenant
<input type="checkbox"/>		データ参照一覧・サンプル1	2013/11/06 18:35	tenant
<input type="checkbox"/>		データ参照一覧・サンプル2(クエリ別	2013/11/06 18:35	tenant

ダイアログが表示されるので、OKボタンをクリックします。

次に、「サイトマップ」→「ポータル - ポートレット一覧」を選択します。

ポータル登録

新規登録

表示先ポータル種別 [ ] アプリケーション [ ] 検索

編集	設定	アプリケーション	名称	説明	アクセ	公開
		イベントサンプル	イベント受信 1 (JSSP)	スクリプト開発モデルポータルレットのイベントサンプルです		<input checked="" type="checkbox"/>
		イベントサンプル	イベント受信 1 (JavaEE)	JavaEE開発モデルポータルレットのイベントサンプルです		<input checked="" type="checkbox"/>
		イベントサンプル	イベント受信 2 (JSSP)	スクリプト開発モデルポータルレットのイベントサンプルです		<input checked="" type="checkbox"/>
		イベントサンプル	イベント受信 2 (JavaEE)	JavaEE開発モデルポータルレットのイベントサンプルです		<input checked="" type="checkbox"/>
		イベントサンプル	イベント送信(JSSP)	スクリプト開発モデルポータルレットのイベントサンプルです		<input checked="" type="checkbox"/>
		イベントサンプル	イベント送信(JavaEE)	JavaEE開発モデルポータルレットのイベントサンプルです		<input checked="" type="checkbox"/>
		インターネット	インターネット	Webクリッピングポータルレットです。		<input checked="" type="checkbox"/>
		ViewCreator	サマリ集計から作成した棒グラフサンプル			<input checked="" type="checkbox"/>

2 ページ中 [ ] ページ目 10 16 件中 1 - 10 を表示

アクセス権の設定アイコンを選択すると、ポータルレットに対する認可設定画面が表示されます。

### アクセス権設定

🔍 エクスポート ⚠️ キャッシュクリア

アクションの種類 全てのアクション ▼

リソース	アクション	認証		組織		ロール			
		ゲストユーザ	認証済みユーザ	サンプル会社	その他会社	テナント管理者	認可管理者	メニュー管理者	メニュー適用管理者
ポータル		▼	▼	▼	▼	▼	▼	▼	▼
サマリ集計から作成した棒グラフ	利用	✖	✖	✖	✖	✔	✖	✖	✖

適宜、適切なアクセス権を設定します。

intra-mart® Top ▼ Workflow ▼ テナント管理 ▼ サンプル ▼ サイトマップ

グループポータル サンプル

#### リンク集

intra-martリンク

- NTTデータイントラマートのホームページ
  - intra-mart FAQ
  - OPEN INTRA-MART

#### サマリ集計から作成した棒グラフサンプル

地域	面積 (面積)	人口 (人口)
埼玉	15,000	5,500
千葉	5,000	4,000
石川	4,500	3,500
滋賀	4,000	1,000
京都	4,800	2,500
兵庫	8,500	4,800
島根	7,000	1,000

登録したポータルをポータルに追加することで表示できます。

ポートレットをポータルに表示する操作については、「[ポータル管理者操作ガイド](#)」を参照してください。

## 物理Viewの作成と表示

作成したクエリをベースにして、データベース上にViewを作成（Create View）できます。

※ViewCreator機能の中でのデータ参照（View）のことではありません。

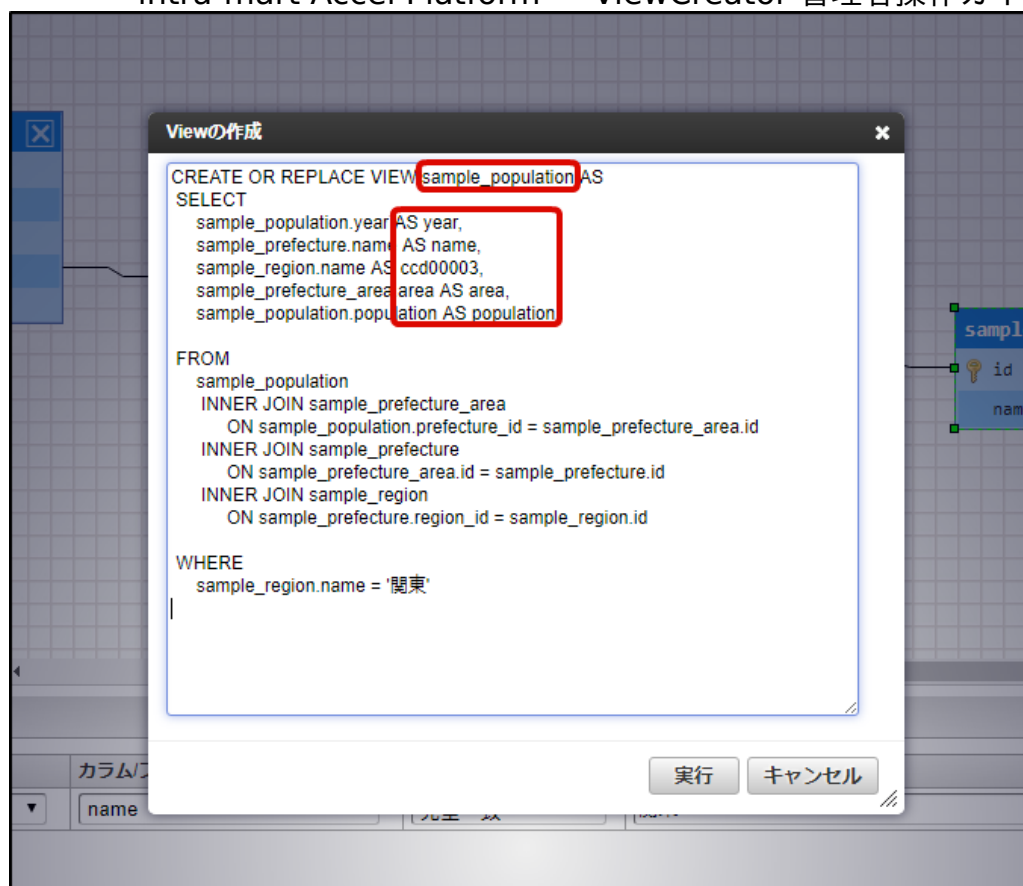
作成したViewはテーブルと同じように、クエリに組み込むことが可能です。

この機能を利用することにより、GUI操作では設定できないような結果を取得する複雑な条件を設定可能です。

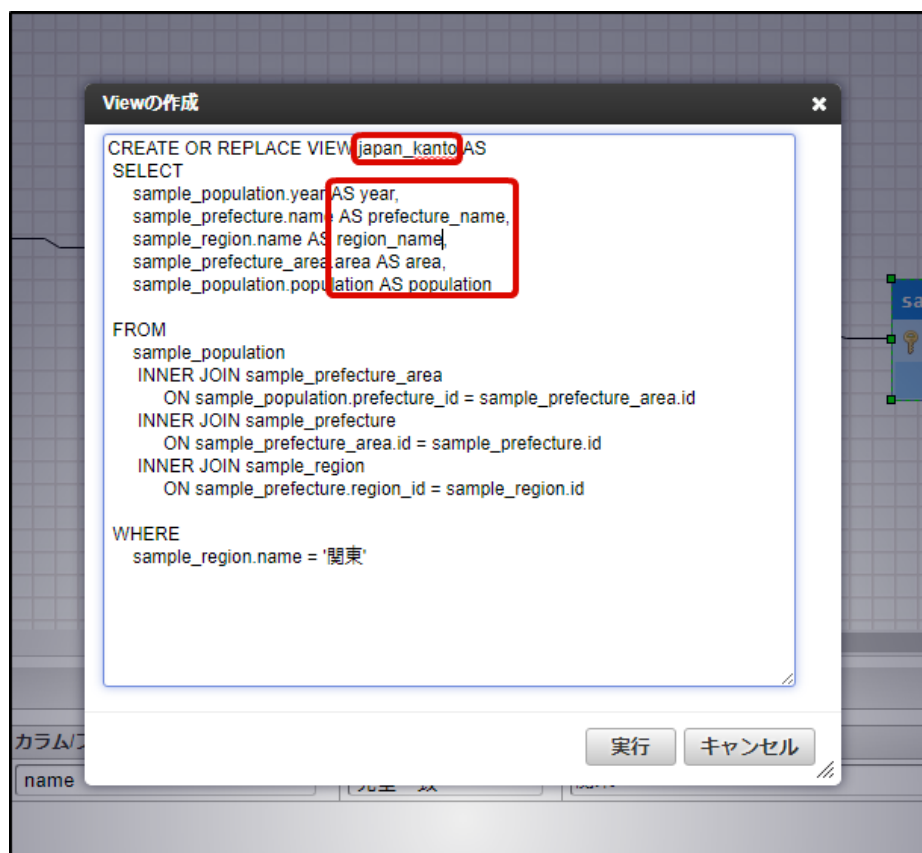
クエリ編集画面のツールバーから、「Viewの作成」をクリックします。

上へ	下へ	テーブルビュー/リソース	カラム/フィールド	型	ソート	キャプション	削除
↑	↓	sample_population	year	数値		年度	✖
↑	↓	sample_prefecture	name	文字列		都道府県	✖
↑	↓	sample_region	name	文字列		地域名	✖

現在編集中のクエリをベースにした、Create View文が自動生成されます。



Viewとフィールドの名称を確認し、適宜SQL文を編集してください。



編集が完了したら「実行」ボタンをクリックします。

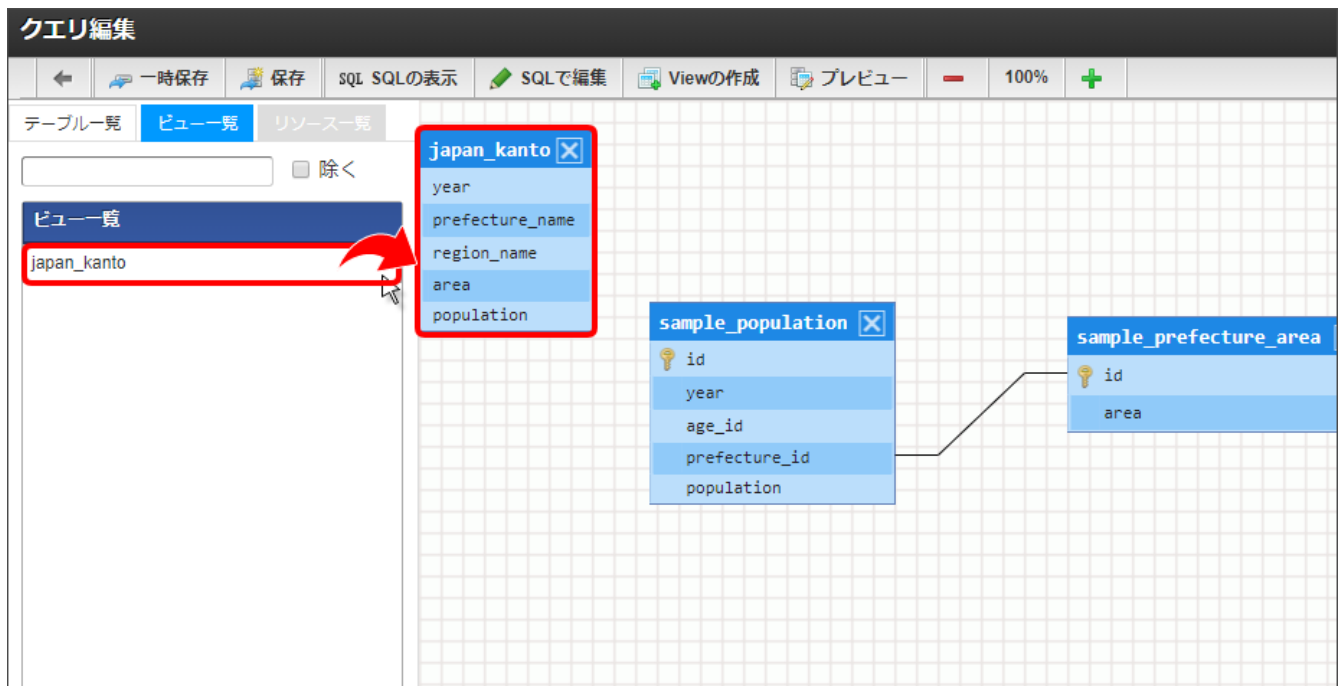


**!** 注意

この機能ではクエリの設定データをベースにSQL文を作成しますが、基本的にどんなSQLでも実行できます。

既存のテーブルやレコードを壊さないように細心の注意が必要です。

例えば、Viewやフィールドの名前に、日本語などの2バイト文字やSQL文で特殊な意味を持つ文字（' など）は使用を避けてください。



正常に登録完了すると、「ビュー」タブの一覧に作成したViewが追加されます。  
作成したViewはテーブルと同じようにダブルクリックでクエリに追加できます。

**i** コラム

物理Viewの作成はクエリビルダでのみ利用可能です。  
SQLビルダでは利用できません。

## 国際化とヘッダ・フッタ

データ参照に付加的に説明などを表示したい場合には、ヘッダおよびフッタの設定を利用してください。  
また、ヘッダ・フッタとデータ参照名は言語ごとに表示を変えることが可能です。

- データ参照名の国際化

「国際化データ」をクリックして、言語ごとの表示名を入力します。

データ参照名\* デフォルト設定

年度 - 地域 - 都道府県 サマリ集計

▼ 国際化データ

英語	Years-Regions-Prefectures summary
日本語	年度 - 地域 - 都道府県 サマリ集計
中国語 (中華人民共和国)	年度 - 区域 - 都道府県 統計

クエリ名 日本でのデータ

集計パターン サマリ ▼

説明

エクスポート機能の利用  利用する

表示設定

枠線  枠線を表示する

国際化データが未入力の場合は、デフォルトの設定が適用されます。

ヘッダとフッタも同様に言語ごとに設定を変えることができます。

ページ設定

1ページに表示する行数 50

ヘッダー

デフォルト設定

▼ 国際化データ

英語

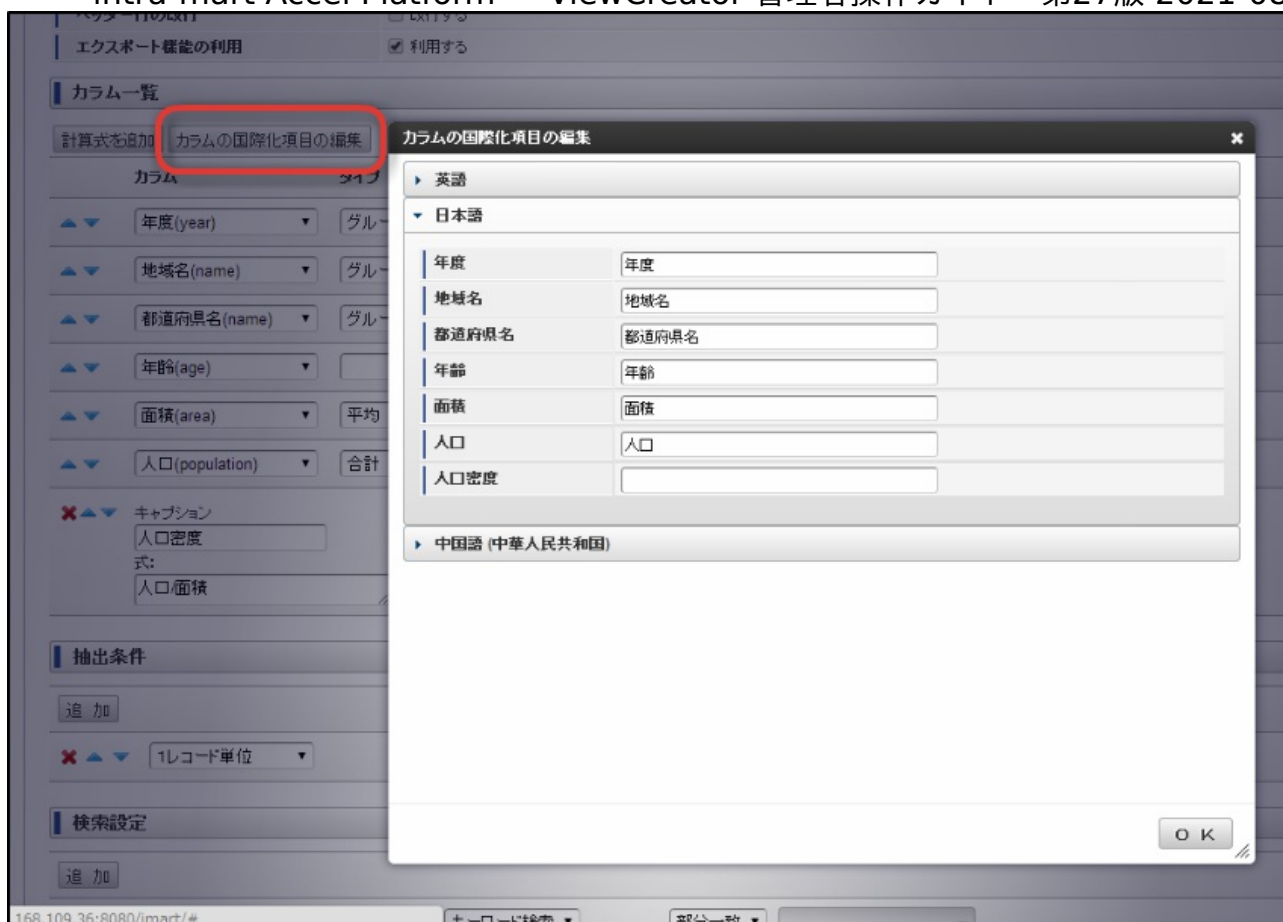
日本語

ヘッダとフッタはリッチテキストが利用可能です。

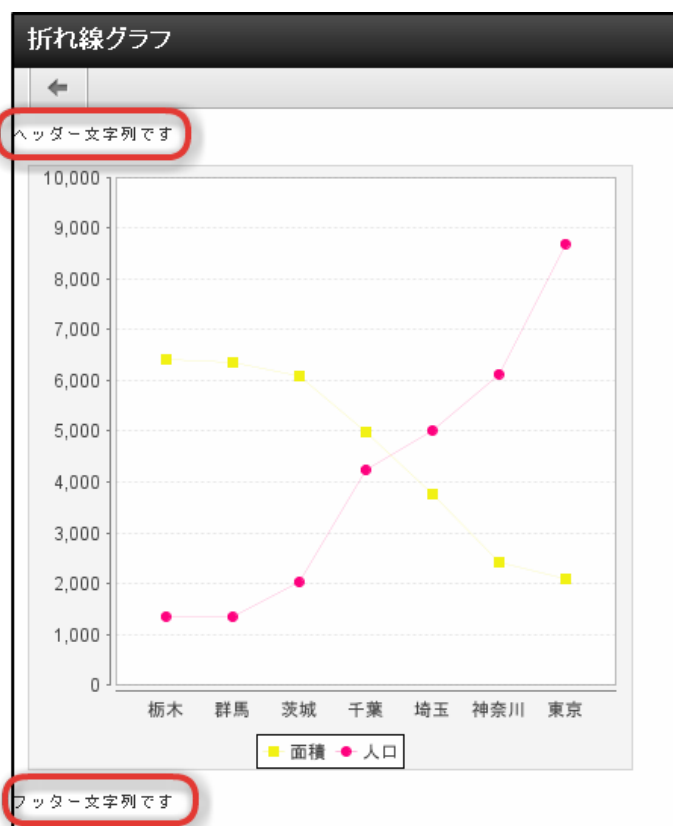
### コラム

リッチテキストは2014 Spring(Granada)から利用可能です。

コラムヘッダの表示名については、「コラムの国際化項目の編集」ボタンで表示されるダイアログ上で入力します。



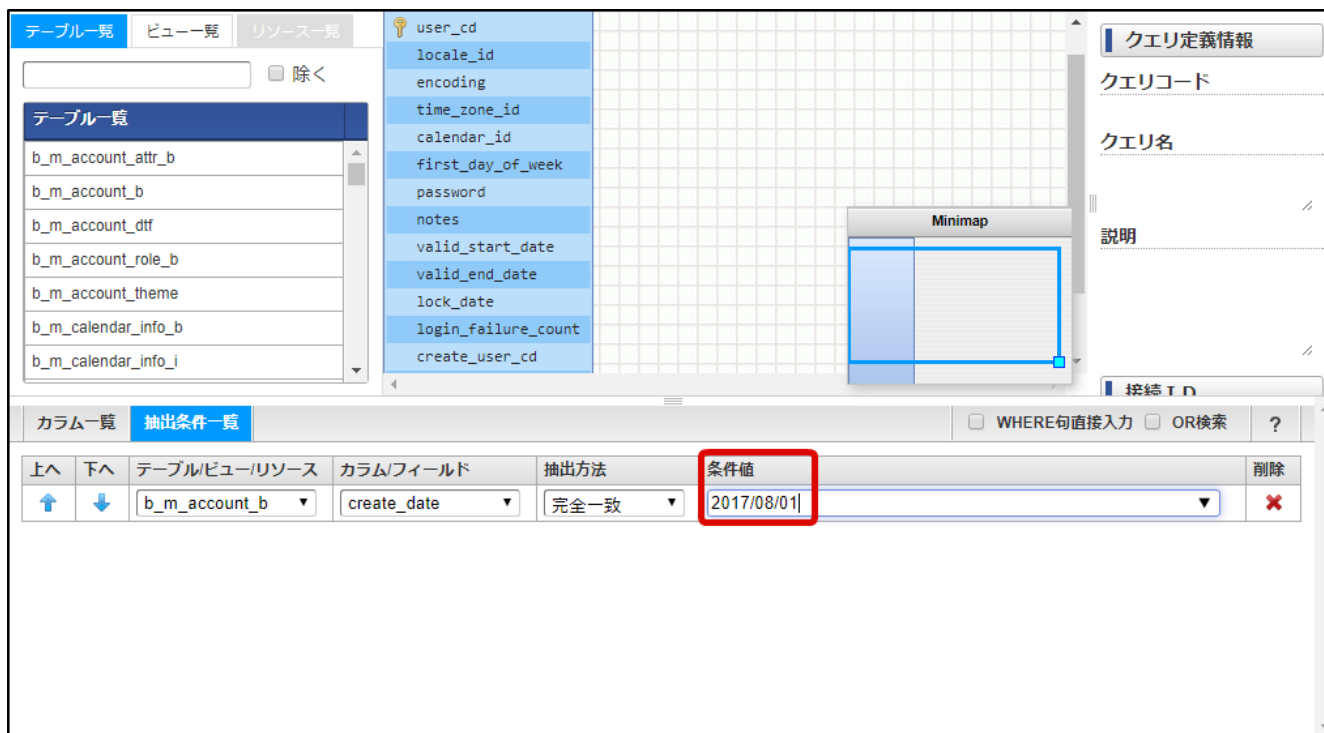
■ 表示例



日付フォーマット

ViewCreatorではタイムスタンプ型や日付型の抽出条件や検索設定を扱うことができます。

条件値は「フォーマットパターン」に沿った形式で入力する必要があります。



図：条件値設定例 - クエリ編集画面

ViewCreatorで利用されるフォーマットパターンは以下のファイルで設定しており、追加することも可能です。

- %コンテキストパス%/WEB-INF/conf/viewcreator-config.xml

#### 【設定例】

```
<date-format-list>
  <date-format>yyyy/MM/dd HH:mm:ss</date-format>
  <date-format>yyyy/MM/dd HH:mm</date-format>
  <date-format>yyyy/MM/dd</date-format>
  <date-format>yyyy/MM</date-format>
</date-format-list>
```

設定されたフォーマットパターンは上から順に優先して適用されますが、条件値が該当しない場合には、DateTimeFormatterクラスに準拠したフォーマットパターンが利用されます。

#### 注意

この設定はデータ参照を閲覧したときの、検索設定の設定値にも適用されますので変更を行った場合は、そのことをユーザに通知する必要があります。

#### コラム

変更を反映するにはサーバの再起動が必要です。

## インポート・エクスポート

ViewCreatorのデータは一括でエクスポートおよびインポートを行うことができます。

## 目次

- クエリとデータ参照のインポート/エクスポート
- インポート
  - インポート画面を確認する
  - インポート種別を選択する
  - インポートを実行する
  - インポート結果を確認する
- エクスポート
  - エクスポート画面を確認する
  - エクスポート対象を選択する
  - エクスポートを実行する

## クエリとデータ参照のインポート/エクスポート

ViewCreatorの設定データを一括でエクスポートおよびインポートを行います。

「サイトマップ」→「ViewCreator - クエリとデータ参照のインポート/エクスポート」を選択します。

ViewCreator クエリとデータ参照のインポート/エクスポート

ViewCreatorの設定データを一括でエクスポートおよびインポートします。  
設定データは複数のcsvファイルをzipでまとめてエクスポートされます。

設定データにはルーティング定義のデータは含まれません。  
ViewCreator エクスポート画面でエクスポートしたzipファイルは、本画面でインポートできません。  
本画面でエクスポートしたzipファイルは、ViewCreator インポート画面でインポートできません。

ファイル名       エクスポート      ダウンロード

+ ファイル追加...      × 中断      🗑️ 削除      インポート      アップロード&インポート

設定データ

設定データはzipファイルでエクスポートされます。

### 注意

インポートの際は以下の点に留意してください。

1. エクスポートしたときと同一のデータベース製品、同一のデータベースバージョンを使用してください。
2. エクスポートデータは、intra-mart Accel Platformのバージョン間における下位互換はサポートしません。
  - 新しいバージョンからエクスポートした資材を、古いバージョンにインポートすることはサポートしません。
  - 古いバージョンからエクスポートした資材を、新しいバージョンにインポートすることはサポートします。
3. クエリで設定したデータベース接続IDがインポート先に存在している必要があります。
4. 上記のデータベース上に、クエリで追加したテーブルが全て存在している必要があります。
5. エクスポートデータにポートレットの登録情報は含まれていません。（ポータルでのインポート/エクスポートを利用してください）
6. インポート・エクスポートデータは、UTF-8でエンコードされます。

**!** 注意

- インポート/エクスポートデータには、IM-FormaDesigner for Accel Platform で「ViewCreatorの一覧を利用する」が選択されているアプリケーションのデータも含まれます。
- インポート/エクスポートデータには、ルーティング定義データは含まれません。ルーティング定義をインポート/エクスポートする場合は「[インポート](#)」、「[エクスポート](#)」を使用してください。

**i** コラム

インポート/エクスポート機能は 2021 Spring(Bergamot) からリンク、画面の名称がクエリとデータ参照のインポート/エクスポートに変更されました。

## インポート

「ViewCreator インポート」画面を利用すると、ViewCreatorで作成したルーティング定義をインポートできます。インポート機能を利用する場合、事前に定義をエクスポートする必要があります。エクスポートの方法は「[エクスポート](#)」を参照してください。

## インポート画面を確認する

「サイトマップ」→「ViewCreator - インポート」を選択します。

ViewCreator インポート

エクスポート

**インポート設定**

**i** zip形式で圧縮されたルーティングカテゴリ、ルーティング定義のxmlファイルのインポートを行います。インポート種別が新規の場合、新規登録を行います。登録対象と同じキーのデータが登録されている場合は登録をスキップします。インポート種別が更新の場合、更新登録を行います。更新対象のデータが存在しない場合は新規登録を行います。

**!** ViewCreator クエリとデータ参照のインポート/エクスポート画面でエクスポートしたzipファイルは、本画面でインポートできません。

インポート種別\*  新規  更新

インポートファイル\*  選択されていません

## インポート種別を選択する

インポート種別は「新規」または「更新」の2種類が存在します。

- 新規：新規登録を行います。登録対象と同じキーのデータが登録されている場合は登録をスキップします。
- 更新：更新登録を行います。更新対象のデータが存在しない場合は新規登録を行います。

## インポートを実行する

1. 「ViewCreator インポート」画面でインポート種別を選択します。
2. 「ファイルを選択」をクリックし、インポートする定義ファイルを選択します。

3. 「インポート実行」をクリックします。

ViewCreator インポート

エクスポート

### インポート設定

zip形式で圧縮されたルーティングカテゴリ、ルーティング定義のxmlファイルのインポートを行います。  
 インポート種別が新規の場合、新規登録を行います。登録対象と同じキーのデータが登録されている場合は登録をスキップします。  
 インポート種別が更新の場合、更新登録を行います。更新対象のデータが存在しない場合は新規登録を行います。

ViewCreator クエリとデータ参照のインポート/エクスポート画面でエクスポートしたzipファイルは、本画面でインポートできません。

インポート種別\*  新規  更新

インポートファイル\*  viewcreator-data.zip

#### インポート結果を確認する

インポートが成功した場合、「ViewCreator インポート結果」画面が表示されます。

- インポート種別に「新規」を選択した場合

インポートした定義情報ごとの総件数、新規登録件数、スキップ件数が表示されます。

スキップ件数が1件以上ある場合、スキップファイルの詳細が表示されます。

スキップファイル詳細の acordeion を押下することで展開され、インポートした定義ごとにファイル名、キー情報、名称が表示されます。

スキップ件数が0の場合

ViewCreator インポート結果

**インポートに成功しました。**

インポート結果

総件数	新規登録件数	スキップ件数
6	6	0

ルーティングカテゴリ

総件数	新規登録件数	スキップ件数
2	2	0

ルーティング定義

総件数	新規登録件数	スキップ件数
4	4	0

インポート画面に戻る

スキップ件数が1件以上の場合

ViewCreator インポート結果

**インポートに成功しました。**

インポート結果

総件数	新規登録件数	スキップ件数
8	2	6

ルーティングカテゴリ

総件数	新規登録件数	スキップ件数
3	1	2

▶ スキップファイル詳細

ルーティング定義

総件数	新規登録件数	スキップ件数
5	1	4

インポート画面に戻る

スキップファイル詳細の acordeion を押下する



ViewCreator インポート結果

### インポート結果

総件数	新規登録件数	スキップ件数
8	2	6

### ルーティングカテゴリ

総件数	新規登録件数	スキップ件数
3	1	2

#### ▼ スキップファイル詳細

ファイル名	カテゴリID	カテゴリ名
category01.xml	category01	category 1 _ja
category02.xml	category02	category 2

### ルーティング定義

総件数	新規登録件数	スキップ件数
5	1	4

#### ▼ スキップファイル詳細

ファイル名	ルーティングID	ルーティング名
route00001.xml	route00001	routing 0 0 0 0 1 _ja
route00002.xml	route00002	routing 0 0 0 0 2 _ja
route00010.xml	route00010	routing 0 0 0 1 0 _ja
route00011.xml	route00011	routing 0 0 0 1 1

[インポート画面に戻る](#)

- インポート種別に「更新」を選択した場合

インポートした定義情報ごとの総件数、新規登録件数、更新件数が表示されます。

ViewCreator インポート結果

インポートに成功しました。

インポート結果

総件数	新規登録件数	更新件数
8	3	5

ルーティングカテゴリ

総件数	新規登録件数	更新件数
3	1	2

ルーティング定義

総件数	新規登録件数	更新件数
5	2	3

インポート画面に戻る

## エクスポート

「ViewCreator エクスポート」画面を利用すると、ViewCreatorで作成したルーティング定義をエクスポートできます。

### エクスポート画面を確認する

「サイトマップ」→「ViewCreator - エクスポート」を選択します。



#### エクスポート対象を選択する

エクスポート対象は「全て」と「選択」の2種類が存在します。

- 全て：確認ダイアログを表示し全てのデータをエクスポートします。
- 選択：エクスポート対象を選択できます。エクスポート実行ボタンを押下すると確認画面を表示します。

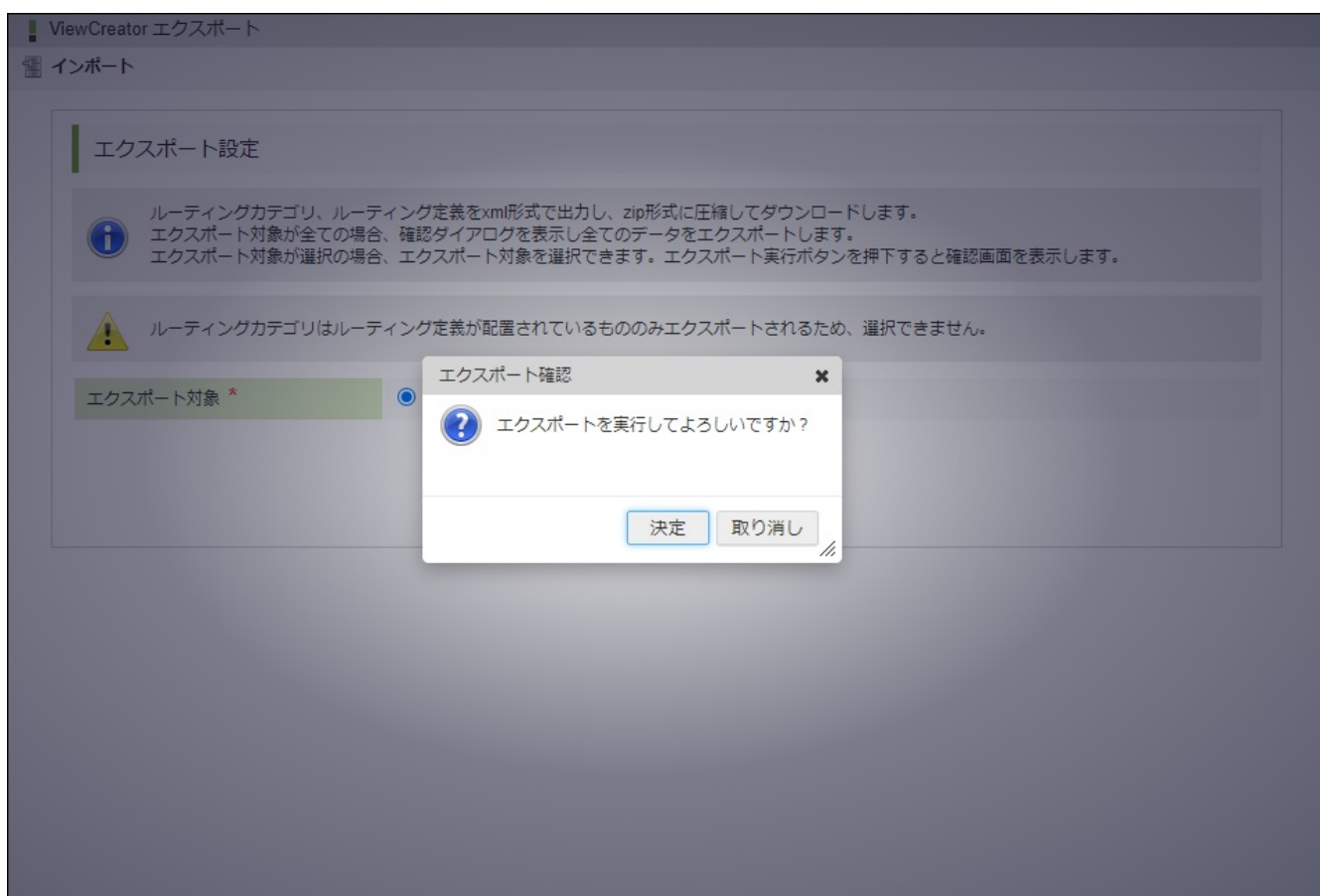
#### エクスポートを実行する

全てのデータをエクスポートする

1. 「ViewCreator エクスポート」画面でエクスポート対象に「全て」を選択します。



2. 「エクスポート実行」をクリックします。
3. 確認ダイアログが表示されるので、「決定」をクリックします。




対象を選択してエクスポートする


1. 「ViewCreator エクスポート」画面でエクスポート対象に「選択」を選択します。

ViewCreator エクスポート

インポート

### エクスポート設定

 ルーティングカテゴリ、ルーティング定義をxml形式で出力し、zip形式に圧縮してダウンロードします。  
 エクスポート対象が全ての場合、確認ダイアログを表示し全てのデータをエクスポートします。  
 エクスポート対象が選択の場合、エクスポート対象を選択できます。エクスポート実行ボタンを押下すると確認画面を表示します。






 ルーティングカテゴリはルーティング定義が配置されているもののみエクスポートされるため、選択できません。

エクスポート対象 \*  全て  選択

### ルーティング定義

選択済み件数 : 0 件

ルーティング名検索

<input type="checkbox"/>	参照	カテゴリID	カテゴリ名	ルーティングID	ルーティング名	クエリコード	クエリ名
<input type="checkbox"/>		category01	category 1	route00001	routing 0 0 0 0 1 _ja	5i7o4o3kli1pj	日本のデータ
<input type="checkbox"/>		category01	category 1	route00002	routing 0 0 0 0 2 _ja	5i7o4o3kli1pj	日本のデータ
<input type="checkbox"/>		category01	category 1	route00003	routing 0 0 0 0 3	5i7o4o3kli1pj	日本のデータ
<input type="checkbox"/>		category02	category 2	route00010	routing 0 0 0 1 0 _ja	5i7o4o3kli1pj	日本のデータ
<input type="checkbox"/>		category02	category 2	route00011	routing 0 0 0 1 1	5i7o4o3kli1pj	日本のデータ

1 << 1ページ中 1 ページ目 >> 50 5件中 1 - 5 を表示


エクスポート実行


- ルーティング定義からエクスポートする定義にチェックをつけ、「エクスポート実行」をクリックします。

ViewCreator エクスポート

インポート

### エクスポート設定

 ルーティングカテゴリ、ルーティング定義をxml形式で出力し、zip形式に圧縮してダウンロードします。  
エクスポート対象が全ての場合、確認ダイアログを表示し全てのデータをエクスポートします。  
エクスポート対象が選択の場合、エクスポート対象を選択できます。エクスポート実行ボタンを押下すると確認画面を表示します。






 ルーティングカテゴリはルーティング定義が配置されているもののみエクスポートされるため、選択できません。

エクスポート対象 \*  全て  選択

### ルーティング定義

選択済み件数 : 2 件


ルーティング名検索

	参照	カテゴリID	カテゴリ名	ルーティングID	ルーティング名	クエリコード	クエリ名
<input checked="" type="checkbox"/>		category01	category 1	route00001	routing 0 0 0 0 1 _ja	5i7o4o3kli1pj	日本のデータ
<input type="checkbox"/>		category01	category 1	route00002	routing 0 0 0 0 2 _ja	5i7o4o3kli1pj	日本のデータ
<input type="checkbox"/>		category01	category 1	route00003	routing 0 0 0 0 3	5i7o4o3kli1pj	日本のデータ
<input checked="" type="checkbox"/>		category02	category 2	route00010	routing 0 0 0 1 0 _ja	5i7o4o3kli1pj	日本のデータ
<input type="checkbox"/>		category02	category 2	route00011	routing 0 0 0 1 1	5i7o4o3kli1pj	日本のデータ

1 << 1ページ中 1 ページ目 >> 50 5件中 1 - 5 を表示

エクスポート実行

3. 「ViewCreator エクスポート対象確認」画面が表示されるので、エクスポート対象を確認し「決定」をクリックします。

 以下の情報をエクスポートします。よろしいですか？

▼ ルーティング定義					選択済み件数：2件
カテゴリID	カテゴリ名	ルーティングID	ルーティング名	クエリコード	クエリ名
category01	category 1	route00001	routing 0 0 0 0 1_ja	5i7o4o3kli1pj	日本のデータ
category02	category 2	route00010	routing 0 0 0 1 0_ja	5i7o4o3kli1pj	日本のデータ

決定

キャンセル



#### 注意

ルーティングカテゴリはルーティング定義が配置されているもののみエクスポートされるため、選択できません。

## テーブル名とフィールド名のキャプション表示

クエリ編集画面のテーブル一覧やフィールド一覧は、キャプションを表示させることができます。

クエリ編集

← 一時保存 保存 SQL SQLの表示 SQLで編集 Viewの作成 プレビュー 100% +

テーブル一覧 ビュー一覧 リソース一覧

検索  除く

テーブル一覧
会社(imm_company)
会社グループ(imm_company_grp)
会社グループ所属(imm_company_grp_ath)
会社グループ内包(imm_company_grp_inc_a)
会社グループセット(imm_company_grp_set)
役職(imm_company_post)
法人(imm_corporation)
法人所属(imm_corporation_ath)
法人分類(imm_corporation_ctg)
法人分類所属(imm_corporation_ctg_ath)
法人分類項目(imm_corporation_ctg_itm)
法人グループ(imm_corporation_grp)
法人グループ所属(imm_corporation_grp_ath)
法人グループ内包(imm_corporation_grp_inc)
法人グループセット(imm_corporation_grp_s)
通貨(imm_currency)
通貨換算コード(imm_currency_conversion)

会社(imm\_company) [X]

- 会社コード(company\_cd)
- ソートキー(sort\_key)
- 作成者(create\_user\_cd)
- 作成日(create\_date)
- 最終更新者(record\_user\_cd)
- 最終更新日(record\_date)

詳しい設定方法については、「[テーブルメンテナンス管理者ガイド - テーブルの拡張情報の設定](#)」を参照してください。

## 外部データソースの利用

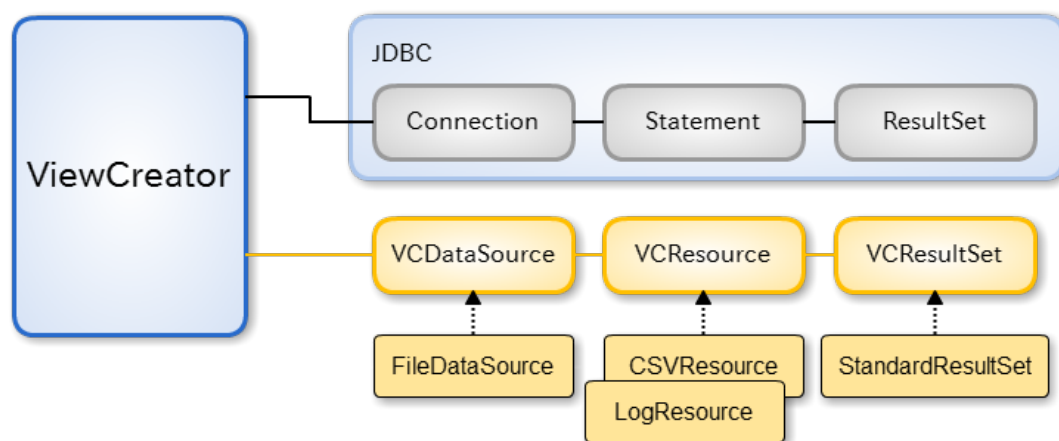
ViewCreatorでは、RDB（リレーショナルデータベース）以外に格納されたデータを扱うためのインターフェースを用意しています。標準では静的なファイル（CSVファイルおよび、ログファイル）を参照するための実装を提供しています。

### 目次

- インタフェース一覧
- CSVファイルおよびログファイルのリソース登録
  - リソースの設定項目
  - ファイルリソースとRDBの連携
- ロジックフローの利用
  - ViewCreatorで利用可能なロジックフローの出力設定
  - 「ロジックフロー管理」画面
  - ロジックフローを検索する
  - クエリの作成

## インタフェース一覧





#### ■ VCDDataSourceインタフェース

このインタフェースの実装は、複数のVCRResourceインスタンスへのアクセスを提供します。  
新しい外部データソースの設定は、%コンテキストパス%/WEB-INF/conf/viewcreator-config.xmlで行います。

##### ■ FileDataSourceクラス

静的なファイルリソースへのアクセスを提供するクラスです。

#### ■ VCRResourceインタフェース

VCRResourceはあらかじめ決められたフォーマットの2次元データを表します。

##### ■ CSVResourceクラス

CSVファイルをリソースとして扱います。

##### ■ LogResourceクラス

ログファイル(\*.log)をリソースとして扱います。

#### ■ VCResultSetインタフェース

リソースから取得した結果データセットを表します。

### **i** コラム

インタフェースの詳細についてはAPIリストを参照してください。

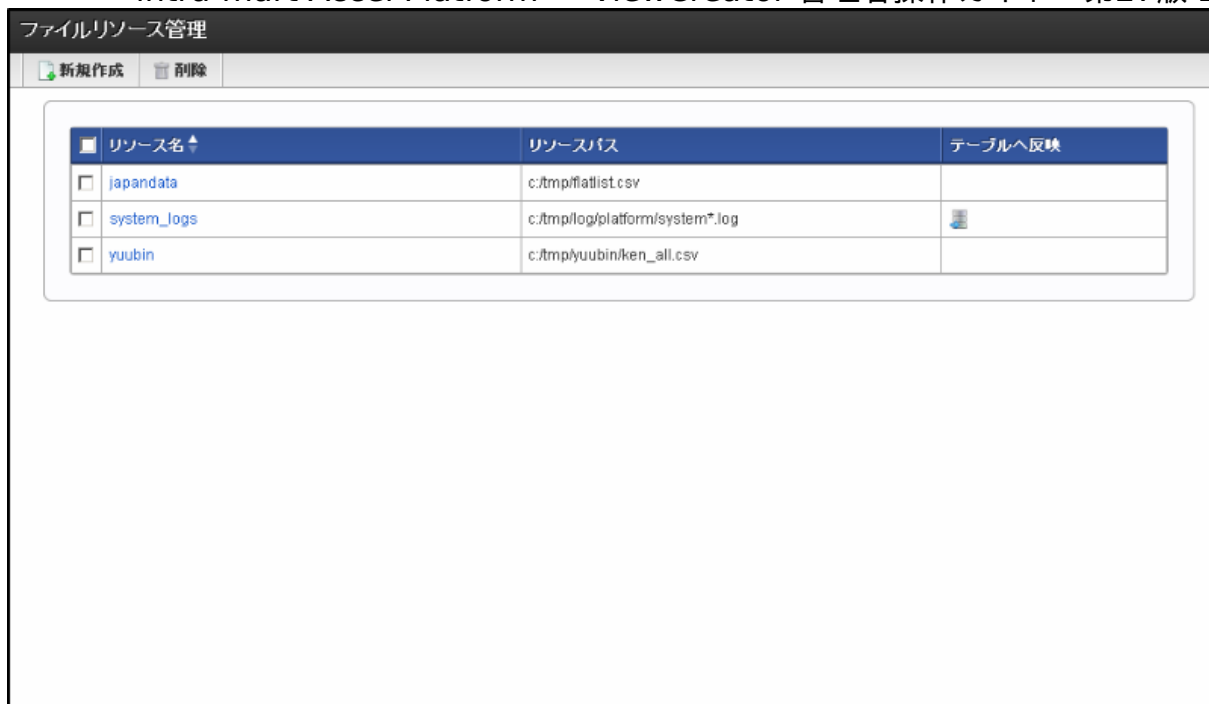
```

jp.co.intra_mart.foundation.viewcreator.expansion.VCDDataSource
jp.co.intra_mart.foundation.viewcreator.expansion.VCRResource
jp.co.intra_mart.foundation.viewcreator.expansion.VCResultSet
  
```

## CSVファイルおよびログファイルのリソース登録

テキストファイル（CSVファイルおよびログファイル）の場合は、プログラムを書くことなく簡単な設定をするだけでViewCreator上で扱うことができます。

「サイトマップ」→「ViewCreator」→「ファイルリソース管理」をクリックします。



新規登録をクリックすると、下記のようなダイアログが表示されます。



## リソースの設定項目

項目名	説明
リソース名	ファイルリソースの名称です。半角英数字およびアンダースコアのみ入力可能です。 ※RDBにおけるテーブル名に相当します。
エンコーディング	テキストファイルのエンコーディングを指定します。デフォルトは「UTF-8」です。

項目名	説明
リソースパス	<p>ファイルリソースのパスを入力します。相対パスが入力された場合は、ストレージディレクトリ内を検索します。</p> <p>またファイル名の一部に「*（アスタリスク）」を入力することで、複数のファイルを指定することも可能です。</p> <p>例) system/platform/system*.log</p> <p>その場合、対象となるファイルのフォーマットはすべて同じである必要があります。</p>
リソースクラス名	<p>オリジナルのリソースクラスを使用する場合にクラス名を入力します。</p> <p>デフォルトでは、リソースファイルの拡張子が「.log」のファイルはLogResourceクラスが適用され、それ以外のファイルはCSVResourceクラスが適用されます。</p> <p>クラス名を入力する場合は、VCResourceを実装している必要があります。</p>
データベースとの連携	<p>この設定項目にチェックが付けられた場合、テナントデータベースに新しくテーブルが作成されます。</p> <p>テキストファイルの内容は、作成されたテーブルにインポートされます。</p> <p>※テーブル名は、「imvc_rc_」 + リソース名です。</p> <p>また、この設定を有効にするにはリソースクラスが下記のクラスのサブクラスである必要があります。</p> <p>jp.co.intra_mart.foundation.viewcreator.expansion.impl.ResourceWithTable</p> <p>※標準のCSVResourceクラスとLogResourceクラスは上記クラスのサブクラスです。</p>
セパレータ	<p>データの区切り文字を指定します。</p> <p>設定が省略された場合、</p> <p>CSVResourceクラスのデフォルト値は、「,（カンマ）」</p> <p>LogResourceクラスのデフォルト値は、「\t（TAB）」です。</p>
日付フォーマット	<p>後述のフィールドのデータ型で「タイムスタンプ」が指定されたフィールドデータのフォーマットパターンを指定します。</p>
最大グループ数	<p>このリソースを使用して「リスト集計」が作成された場合の、「グループ化」列の最大項目数を指定します。</p> <p>OutOfMemoryErrorを起こさないようにするための設定です。</p> <p>省略時のデフォルト値は20000です。</p>
フィールド一覧	<p>テキストデータのデータ構成を設定します。1行分のデータをセパレータで区切った要素単位でフィールドを設定します。</p> <p>データ型は、データの性質に合わせて「文字列、数値、タイムスタンプ」の中から設定します。</p> <p>タイムスタンプが設定されたフィールドは、日付フォーマットの設定値を用いて解析が行われません。</p>

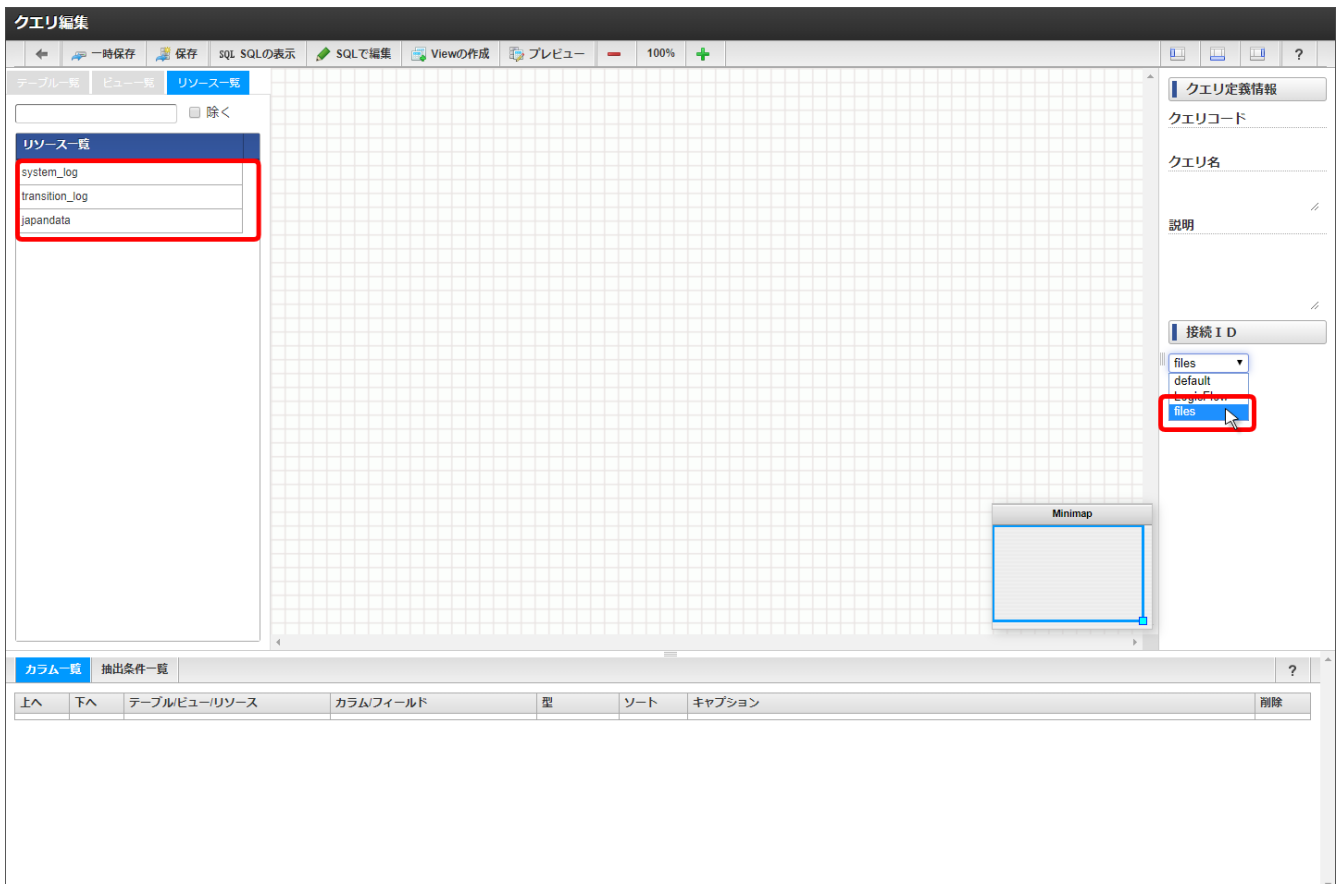
### 注意

標準のCSVResourceクラスとLogResourceクラスはソートの実装を提供しません。  
従って、これらの標準クラスを使用して作成されたリスト集計ではソートに関する全ての機能を利用できません。

「プレビュー」ボタンをクリックすると、テキストデータを正しく解析できているか確認できます。



作成したファイルリソースは、クエリ編集画面でRDBのテーブルと同様にクエリに追加できます。  
 接続IDのコンボボックスから「files」を選択すると、テーブル一覧に登録済みのリソースが表示されます。

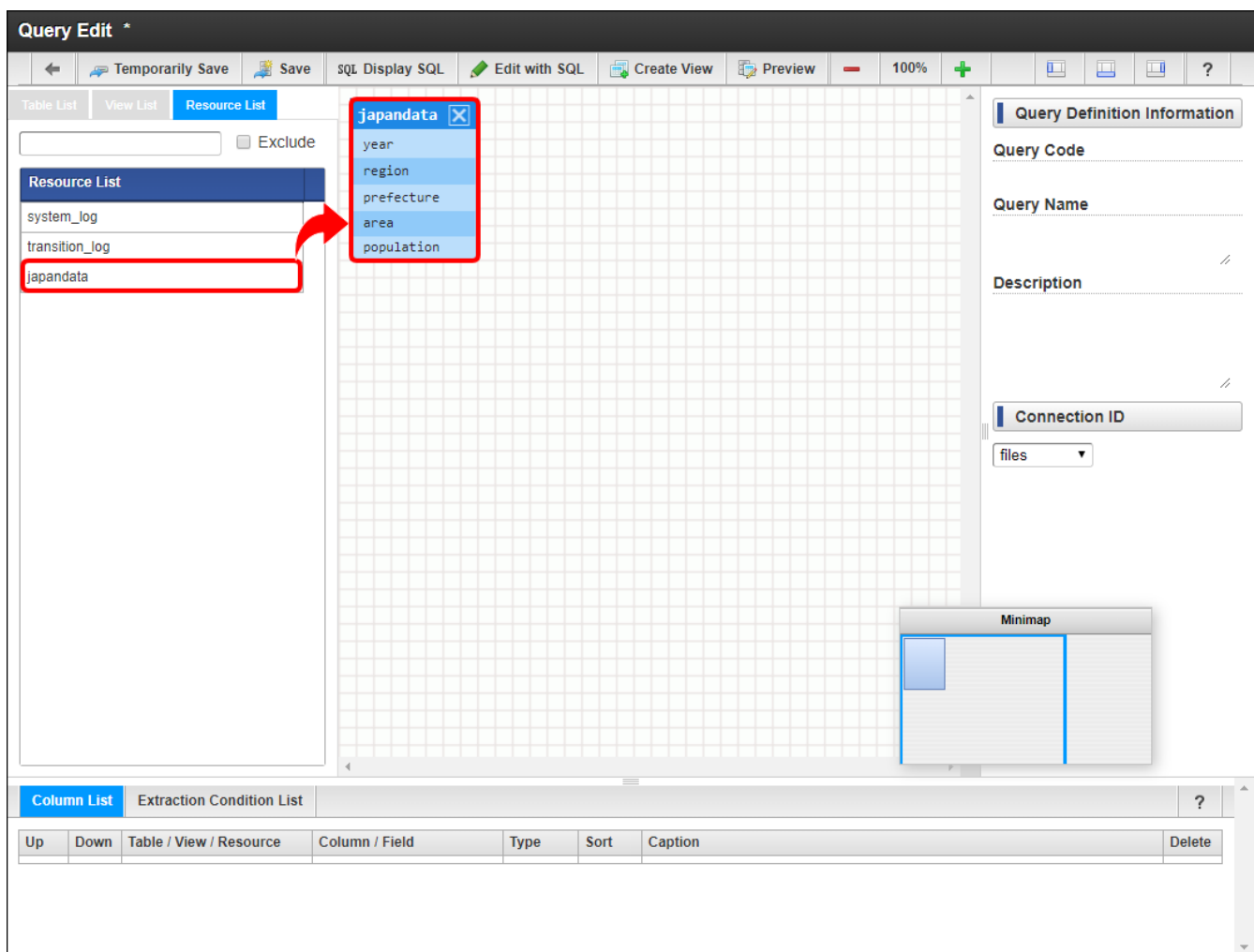


## コラム

接続IDの名前を「files」ではなく別の名前に変更するには、%コンテキストパス%/WEB-INF/conf/viewcreator-config.xmlを編集します。

```
<data-source>
<id>files</id>
<class-name>jp.co.intra_mart.foundation.viewcreator.expansion.impl.FileDataSource</class-name>
</data-source>
```

「リソース」をダブルクリックしてクエリに追加します。



The screenshot shows the 'Query Edit' application. The 'Resource List' tab is selected, displaying a list of resources: system\_log, transition\_log, and japandata. The 'japandata' resource is highlighted with a red box. A red arrow points from the 'japandata' resource to a pop-up window showing its columns: year, region, prefecture, area, and population. The 'Connection ID' is set to 'files'. A 'Minimap' window is also visible.

コラムの追加や抽出条件の設定もRDB上のテーブルと同じように行います。

## 注意

外部データソースを使用するクエリは、以下の制限が適用されます。

- ・クエリに追加できるリソースは1つのみです (JOIN等は一切行えません)。
- ・SQLの生成はできません。
- ・抽出条件で、WHERE句の直接入力はできません
- ・SQLビルダでは外部データソースは利用できません。

## ファイルリソースとRDBの連携

テキストファイルのデータ量が大きい場合、またはJOINや複雑な抽出条件を設定したい場合は

新しくテーブルを作成して、そのテーブルへテキストファイルの内容を反映させることができます。

ファイルリソース登録の際に、「データベースとの連携」にチェックを入れて登録します。

リソースパス	c:/tmp/fatlist.csv								
リソースクラス名									
データベースとの連携	<input checked="" type="checkbox"/> テーブルを作成してデータ連動する								
セパレータ									
日付フォーマット	[YYYY-MM-DD HH:mm:ss.SSS]								
最大グループ数									
フィールド一覧	<table border="1"> <thead> <tr> <th>フィールド名</th> <th>データ型</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/> year</td> <td>数値</td> </tr> <tr> <td><input type="checkbox"/> region</td> <td>文字列</td> </tr> <tr> <td><input type="checkbox"/> prefecture</td> <td>文字列</td> </tr> </tbody> </table>	フィールド名	データ型	<input type="checkbox"/> year	数値	<input type="checkbox"/> region	文字列	<input type="checkbox"/> prefecture	文字列
フィールド名	データ型								
<input type="checkbox"/> year	数値								
<input type="checkbox"/> region	文字列								
<input type="checkbox"/> prefecture	文字列								

### 注意

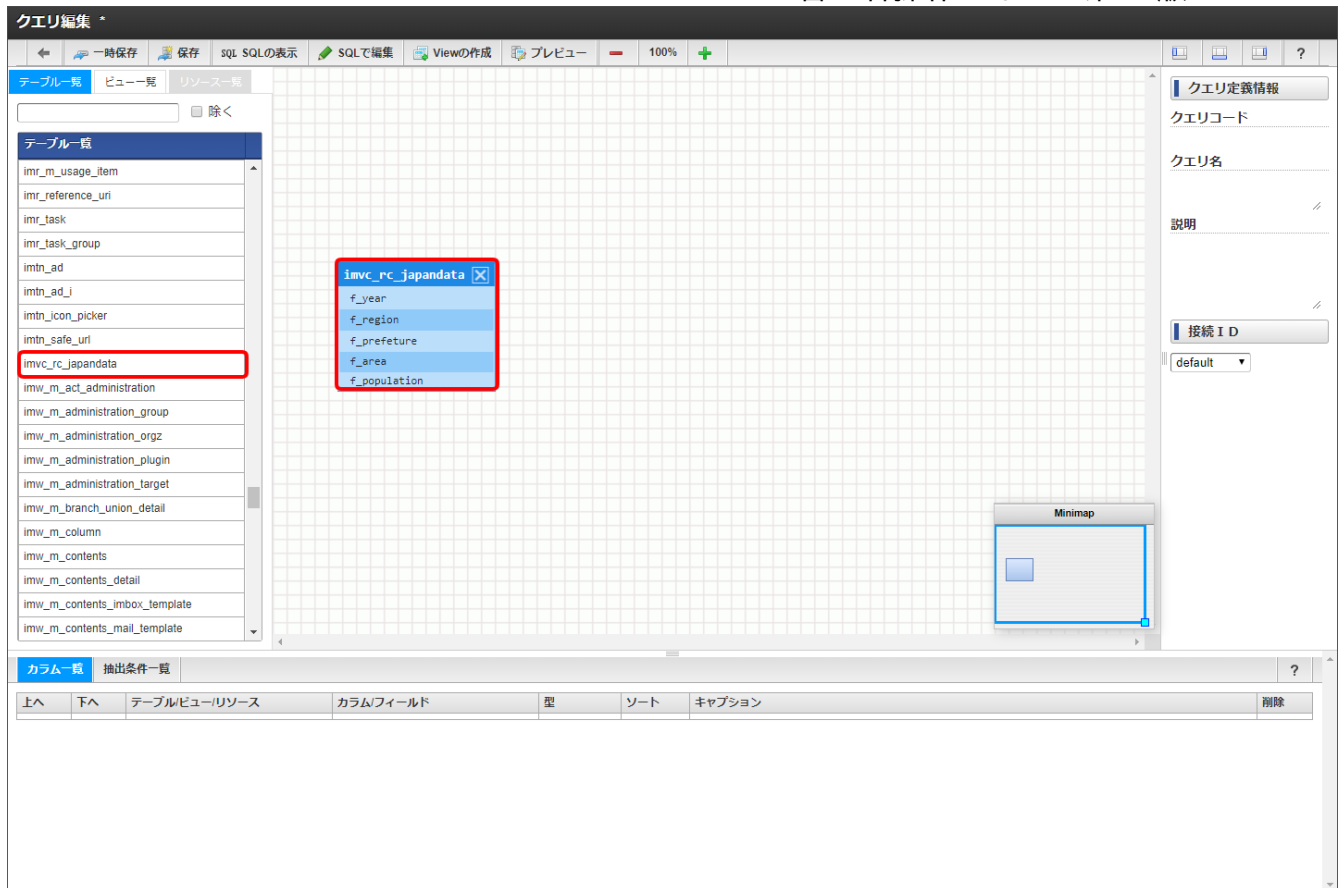
このチェックを入れて登録処理を行った場合、テーブルの作成と同時にデータのインポートも同時に実行されます。データ量によっては処理が完了するまでに時間がかかる場合があります。

ファイルリソースの登録後、テキストファイルのデータ内容に変更があった場合は

「テーブルへ反映」アイコンをクリックすることで、テキストファイルの内容をテーブルへ反映できます。

リソース名	リソースパス	テーブルへ反映
<input type="checkbox"/> japandata	c:/tmp/fatlist.csv	
<input type="checkbox"/> system_logs	c:/tmp/log/platform/system*.log	
<input type="checkbox"/> yuubin	c:/tmp/yuubin/ken_all.csv	

テーブルはテナントデータベース上に作成されます。



テーブル名には imvc\_rc、フィールド名には f\_ がプレフィックスとして付加されます。

### 注意

ファイルリソースの設定の削除時に、テーブルの削除（ドロップ）は行われません。  
必要に応じて手動で削除してください。

## ロジックフローの利用

IM-LogicDesigner のロジックフローを外部データソースとして利用できます。

「[ロジックフロー管理](#)」画面で、外部データソースとして利用したいフローを登録できます。

### コラム

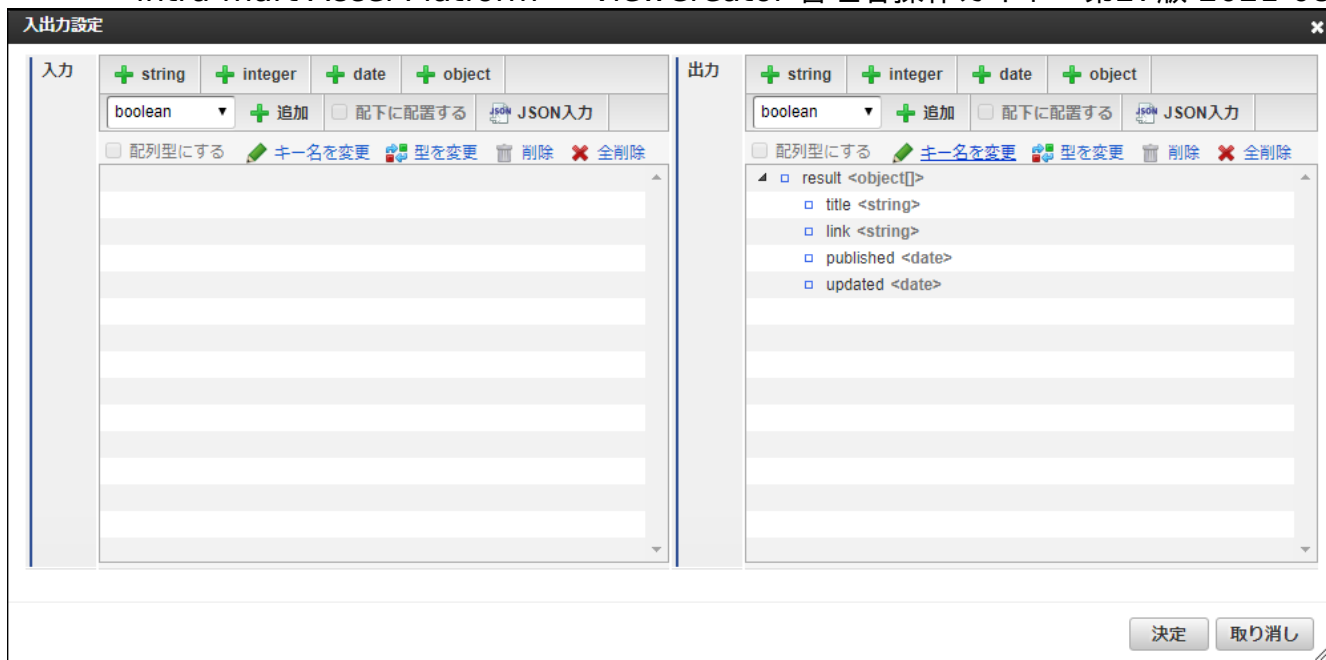
ロジックフローの利用は 2017 Winter(Rebecca) から利用可能です。

ロジックフローを外部データソースとして利用するには、ViewCreatorに利用可能な出力設定を定義する必要があります。  
詳細については「[ViewCreatorで利用可能なロジックフローの出力設定](#)」を参照してください。

登録した外部データソースを利用するクエリ作成については「[クエリの作成](#)」を参照してください。  
クエリに利用できるリソース名は、半角英数字およびアンダースコアのみです。

## ViewCreatorで利用可能なロジックフローの出力設定

IM-LogicDesigner のフロー定義編集において、ViewCreatorに対応した出力設定を定義することで、ロジックフローの出力結果をリソースで利用できます。



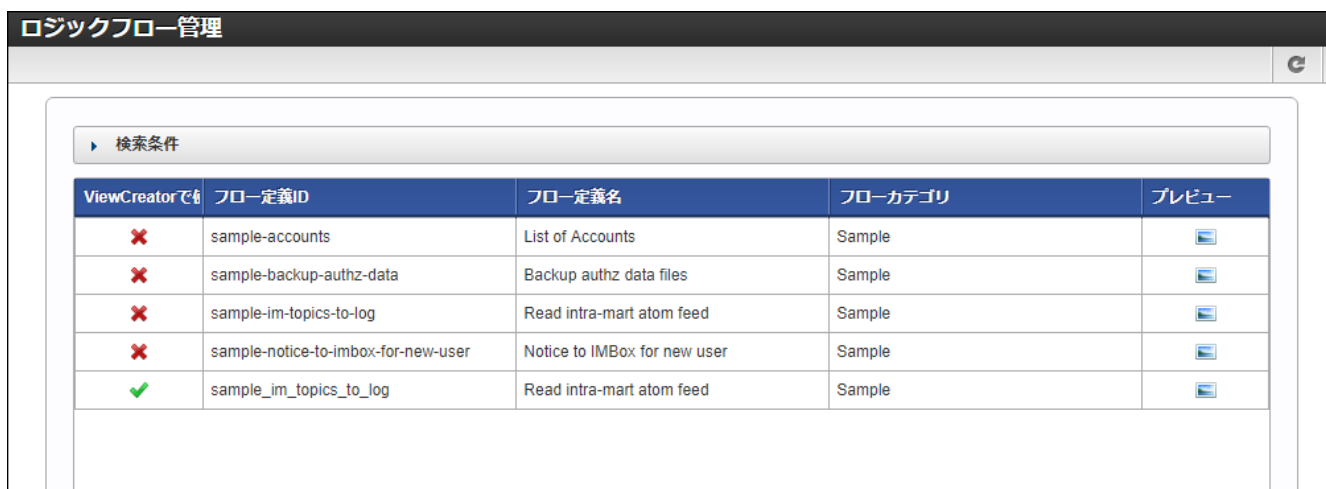
図：「入出力設定」画面

出力設定に「result」というキー名のobject[]型のパラメータを定義します。  
 「result」の配下に配置された各パラメータは、ViewCreatorのリソースの各カラムに利用されます。

### 「ロジックフロー管理」画面

「ロジックフロー管理」画面では、リソースとして利用したいフローを登録できます。

「サイトマップ」→「ViewCreator」→「外部データソース連携」→「ロジックフロー管理」をクリックします。



図：「ロジックフロー管理」画面 - 初期表示

項目名	説明
検索条件	条件を指定し、ロジックフローを検索します。
ViewCreatorで使用する	( <input checked="" type="checkbox"/> ) アイコンをクリックし、( <input checked="" type="checkbox"/> ) にすることでリソースを登録します。 ( <input checked="" type="checkbox"/> ) アイコンをクリックし、( <input checked="" type="checkbox"/> ) にすることでリソースの登録を解除します。
フロー定義ID	ロジックフローのフロー定義IDが表示されます。
フロー定義名	ロジックフローのフロー定義名が表示されます。
フローカテゴリ	ロジックフローのカテゴリが表示されます。
プレビュー	アイコンをクリックすると「ロジックフロー定義のプレビュー」画面が表示されます。

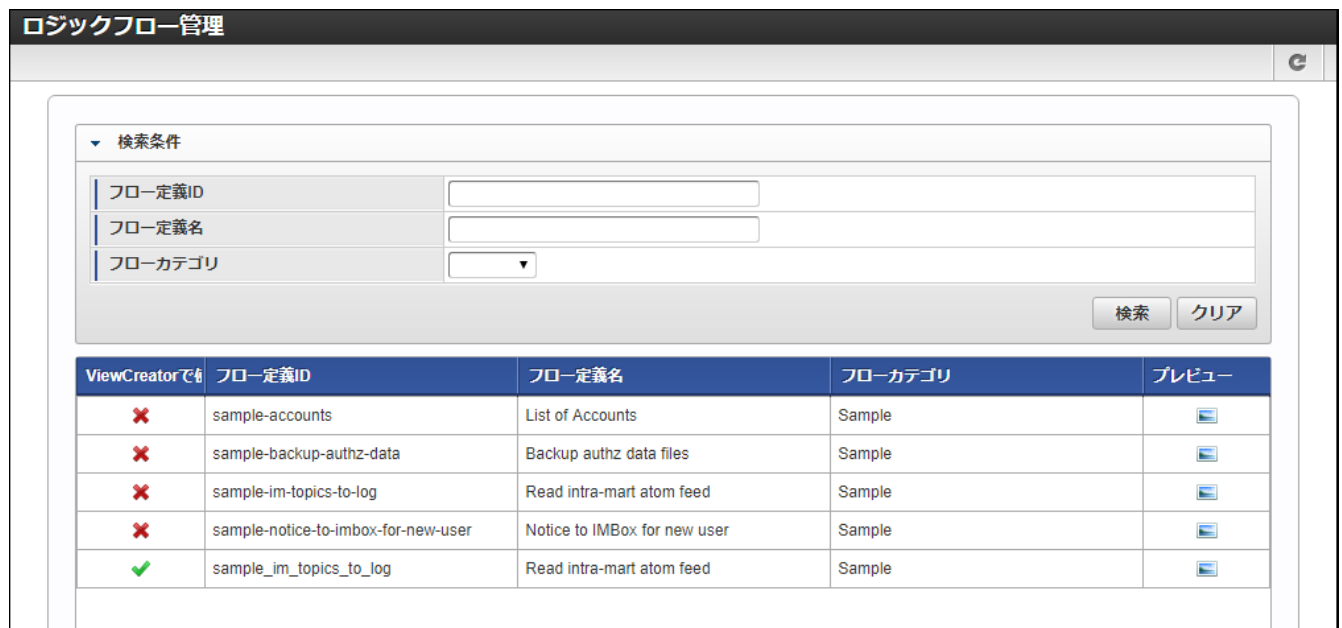
### ロジックフローを検索する



1. 画面中央の「検索条件」をクリックします。



2. 検索フィールドが表示されます。



#### <画面項目>

項目	説明
フロー定義ID	検索するロジックフローのフロー定義IDを表す文字列（の一部）を入力します。
フロー定義名	検索するロジックフローの定義名を表す文字列（の一部）を入力します。
フローカテゴリ	検索するロジックフローが属するフローカテゴリを指定します。
「検索」ボタン	ロジックフローを検索します。
「クリア」ボタン	入力した検索条件をクリアします。



#### コラム

検索フィールドにおけるフローカテゴリのドロップダウンリストは、カテゴリの階層に関係なく、すべてのカテゴリをフラットなリストとして表示します。

1. 検索条件を入力し、「検索」をクリックします。

ロジックフロー管理

検索条件

フロー定義ID

フロー定義名

フローカテゴリ

検索 クリア

ViewCreatorで使	フロー定義ID	フロー定義名	フローカテゴリ	プレビュー
×	sample-accounts	List of Accounts	Sample	
×	sample-backup-Authz-data	Backup authz data files	Sample	
×	sample-im-topics-to-log	Read intra-mart atom feed	Sample	
×	sample-notice-to-imbox-for-new-user	Notice to IMBox for new user	Sample	
✓	sample_im_topics_to_log	Read intra-mart atom feed	Sample	

2. 検索結果が表示されます。

ロジックフロー管理

検索条件

フロー定義ID

フロー定義名

フローカテゴリ

検索 クリア

ViewCreatorで使用する	フロー定義ID	フロー定義名	フローカテゴリ	プレビュー
×	sample-accounts	List of Accounts	Sample	

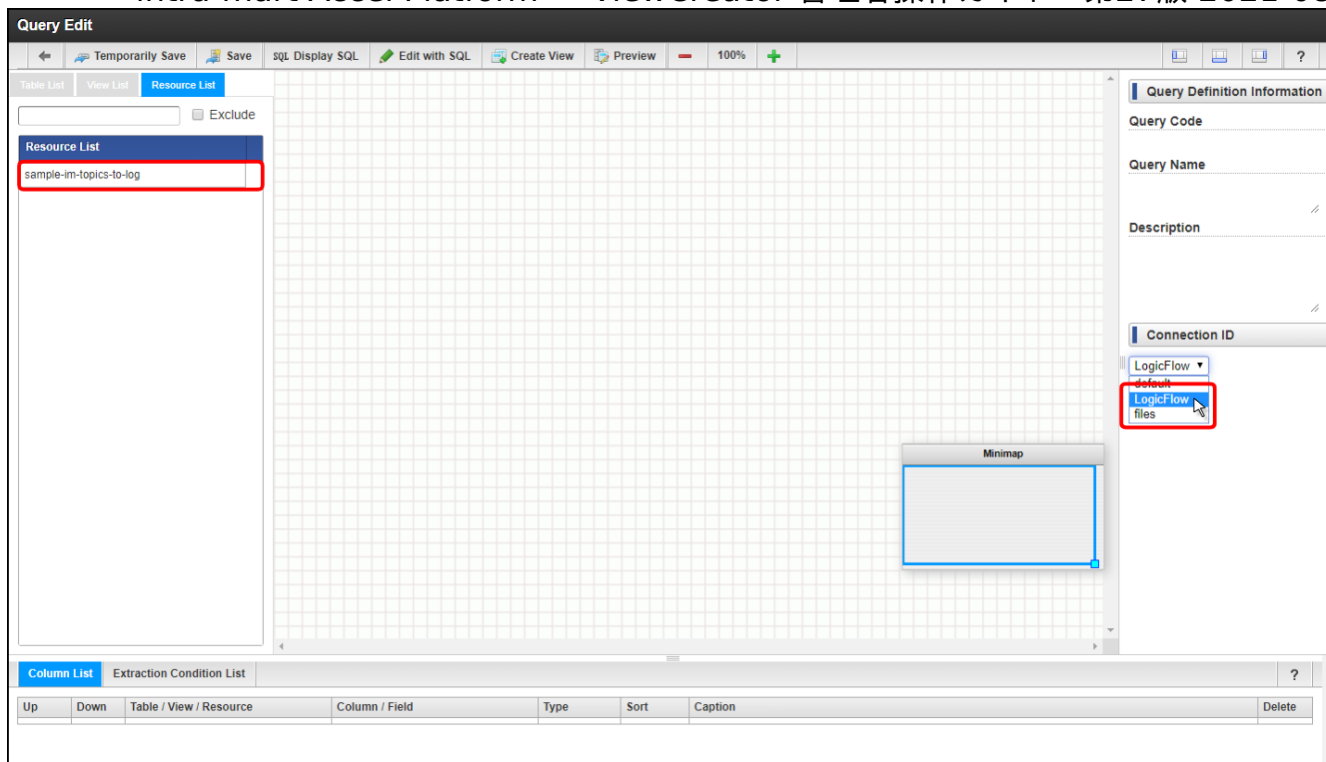


### コラム

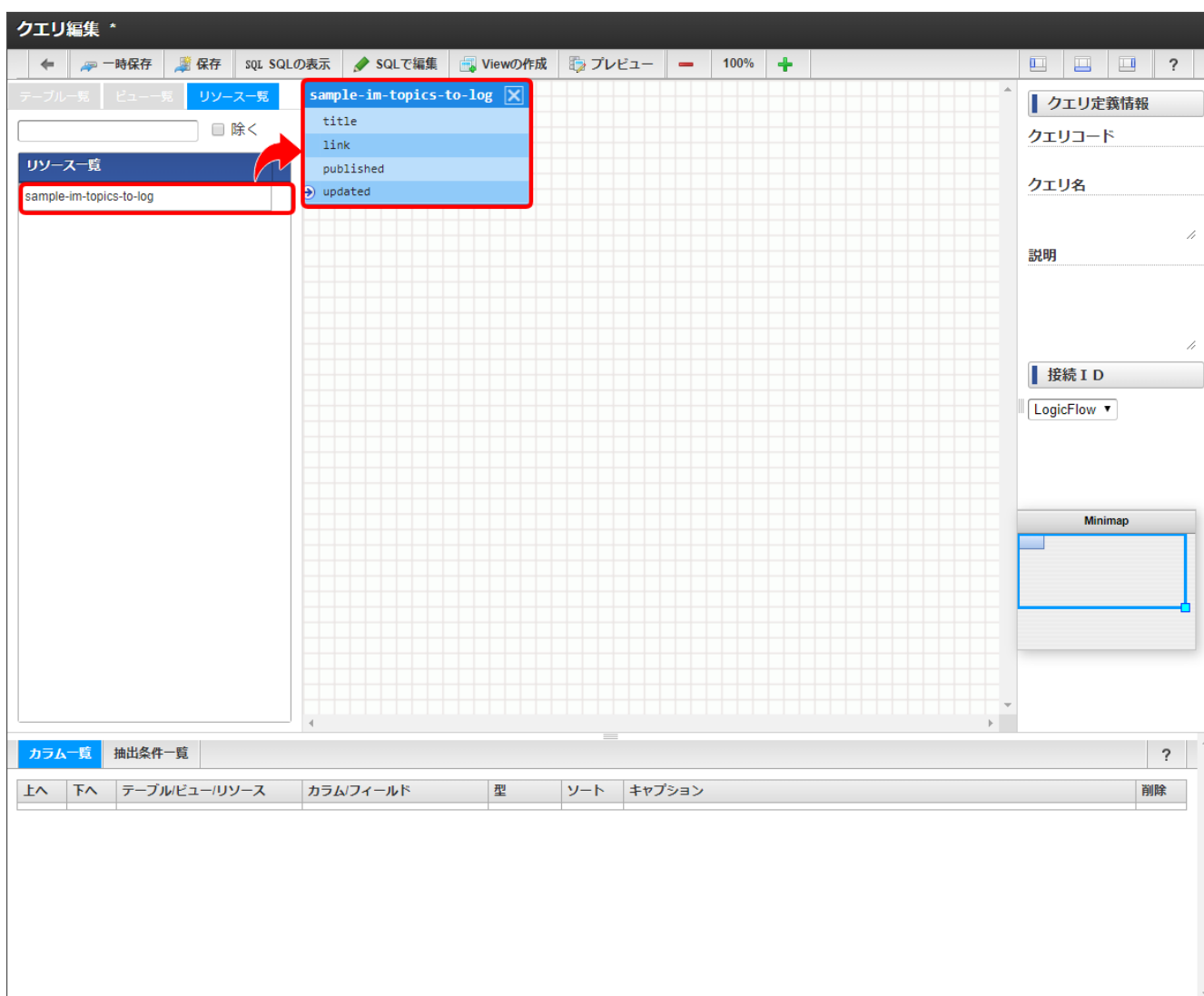
フローカテゴリ選択時の検索結果には、選択されたフローカテゴリの直下に存在するフローのみが表示されます。

## クエリの作成

「ロジックフロー管理」画面で登録したリソースは、クエリ編集画面でRDBのテーブルと同様にクエリに追加できます。接続IDのプルダウンから「logicFlow」を選択すると、テーブル一覧に登録済みのリソースが表示されます。



表示された「リソース」をダブルクリックしてクエリに追加します。



カラムの追加もRDB上のテーブルと同じように行います。

**!** 注意

外部データソースを使用するクエリは、以下の制限が適用されます。

- ・クエリに追加できるリソースは1つのみです（JOIN等は一切行えません）。
- ・SQLの生成はできません。
- ・「抽出条件一覧」は動作しません。「抽出条件一覧」にはフィールドを追加しないでください。
- ・SQLビルダでは外部データソースは利用できません。

## 計算式カラムの設定

リスト集計ではフィールドの値をそのまま表示するだけでなく、計算式や関数を利用できます。

### 目次

- 基本的な使い方
- データ型について
- 式入力ダイアログの利用
- ヘッダとフッタの計算式設定
- リッチテキストと計算式
- 利用可能な関数一覧
- ユーザ定義関数

「計算式を追加」ボタンをクリックすると、リスト集計にカラムが追加されます。

カラム	タイプ	表示	フォーマット	ソート
地域名(name)	グループ化	<input checked="" type="checkbox"/>		<input type="checkbox"/>
都道府県名(name)	グループ化	<input checked="" type="checkbox"/>		<input type="checkbox"/>
年度(year)	グループ化	<input checked="" type="checkbox"/>	#	3桁区切り <input type="checkbox"/>
年齢(age)		<input checked="" type="checkbox"/>		<input type="checkbox"/>
面積(area)	平均	<input type="checkbox"/>	#,###	3桁区切り <input type="checkbox"/>
人口(population)	合計	<input checked="" type="checkbox"/>	#,###	3桁区切り <input type="checkbox"/>
キャプション		<input checked="" type="checkbox"/>	#,##	3桁区切り <input type="checkbox"/>

式:  
人口/面積

### 基本的な使い方

追加された計算式カラムには、以下の2つの入力必須項目があります。

キャプション

カラムの列タイトルとして使用されます。

式:

計算式には他のカラムの「カラム名」を使用できます。データ参照表示時には、該当するカラムのデータが展開されます。

実行結果

近畿のデータ - 1995 リスト

CSV出力

1

年度	地域名	都道府県名	面積	人口	人口密度
1995	近畿	三重	5761	1240	0.215
1995	近畿	滋賀	3855	874	0.227
1995	近畿	京都	4613	1842	0.399
1995	近畿	大阪	1894	6412	3.386
1995	近畿	兵庫	8394	3756	0.447
1995	近畿	奈良	3691	999	0.271
1995	近畿	和歌山	4726	709	0.15

計算式カラム「人口密度」には、「人口 / 面積」の計算結果が表示されています。

計算式カラムの計算結果を別の計算式カラムで使用することもできます。

カラム一覧

計算式を追加 カラムの国際化項目の編集

カラム	タイプ	表示	フォーマット	ソート順	パラメータ
▲ ▼ 地域名(name)	グループ化 ▼	<input checked="" type="checkbox"/>		▼	
▲ ▼ 都道府県名(name)	グループ化 ▼	<input checked="" type="checkbox"/>		▼	
▲ ▼ 年度(year)	グループ化 ▼	<input checked="" type="checkbox"/>	#	3桁区切り	
▲ ▼ 年齢(age)	▼	<input checked="" type="checkbox"/>		▼	
▲ ▼ 面積(area)	平均 ▼	<input type="checkbox"/>	#,###	3桁区切り	
▲ ▼ 人口(population)	合計 ▼	<input checked="" type="checkbox"/>	#,###	3桁区切り	
✖ ▲ ▼ キャプション		<input checked="" type="checkbox"/>	###	3桁区切り	
人口密度					
式:					
人口 / 面積					
▲ ▼ キャプション		<input checked="" type="checkbox"/>	#	3桁区切り	
人口密度 + 2					
式:					
人口密度 + 2					

実行結果

## 近畿のデータ - 1995 リスト

年度	地域名	都道府県名	面積	人口	人口密度	人口密度+2
1995	近畿	三重	5761	1240	0.215	2.215
1995	近畿	滋賀	3855	874	0.227	2.227
1995	近畿	京都	4613	1842	0.399	2.399
1995	近畿	大阪	1894	6412	3.386	5.386
1995	近畿	兵庫	8394	3756	0.447	2.447
1995	近畿	奈良	3691	999	0.271	2.271
1995	近畿	和歌山	4726	709	0.15	2.15

 注意

計算式は上から順に実行されます。

下に追加した計算式カラムの結果を上のカラムで使用することはできません。

計算式カラムは式が実行できない場合（式に含まれている内容がグルーピングされているなど）、ブランクで表示されます。

計算式中で使用可能な関数が用意されています。

<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	キャプション <input type="text" value="人口密度"/>	<input checked="" type="checkbox"/>	<input type="text" value="###"/>	<input type="button" value="3桁区切り"/>
式: <input type="text" value="人口 / 面積"/> <input type="button" value="fx"/>				
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	キャプション <input type="text" value="人口密度+2"/>	<input checked="" type="checkbox"/>	<input type="text" value="#"/>	<input type="button" value="3桁区切り"/>
式: <input type="text" value="人口密度 + 2"/> <input type="button" value="fx"/>				
<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	キャプション <input type="text" value="平均との差(人口)"/>	<input checked="" type="checkbox"/>	<input type="text" value="#"/>	<input type="button" value="3桁区切り"/>
式: <input type="text" value="人口 - AVG(\$人口)"/> <input type="button" value="fx"/>				

AVG関数は引数で指定されたフィールドについて、全レコードの平均値を取得します。

 コラム

“\$”を先頭に付加されたフィールドは、該当レコードの値として展開するのではなくそのフィールドを示す“フィールド名”と解釈されて実行されます。

実行結果

## 近畿のデータ - 1995 リスト

CSV出力

1

年度	地域名	都道府県名	面積	人口	人口密度	人口密度+2	平均との差(人口)
1995	近畿	三重	5761	1240	0.215	2.215	-1,022
1995	近畿	滋賀	3855	874	0.227	2.227	-1,388
1995	近畿	京都	4613	1842	0.399	2.399	-420
1995	近畿	大阪	1894	6412	3.386	5.386	+4,150
1995	近畿	兵庫	8394	3756	0.447	2.447	+1,494
1995	近畿	奈良	3691	999	0.271	2.271	-1,263
1995	近畿	和歌山	4726	709	0.15	2.15	-1,553

## データ型について

計算式の中で利用可能なデータ型は以下の通りです。

- 文字列型
- 数値型
- 日付型
- 真偽値型
- 配列型

レコードから取得したデータの型は、クエリでのカラム定義に従って決定されます。

## クエリで定義された型 計算式で利用した場合の型

文字列 文字列型

数値 数値型

日付 日付型

タイムスタンプ 日付型

真偽値 真偽値型

バイナリ -

- 配列型

## 式入力ダイアログの利用

式は手入力するだけでなく、利用可能な演算子、カラム名（フィールド名）、および関数を一覧から選択しながら作成できます。

関数アイコンをクリックします。

式: 人口/面積

× ▲ ▼ キャプション

式:

抽出条件

ポップアップダイアログには、式の入力エリアの下に利用可能な演算子、カラム、関数の一覧が表示され、クリックすると式の最後に追加挿入できます。



SUM関数をクリックして見ます。



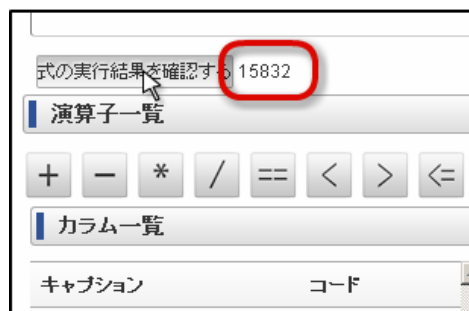


式入力エリアに関数が追加されます。

関数の引数の部分を有効なカラム名に書き換えます。



入力した式の実行結果は「式の実行結果を確認する」ボタンで確認できます。



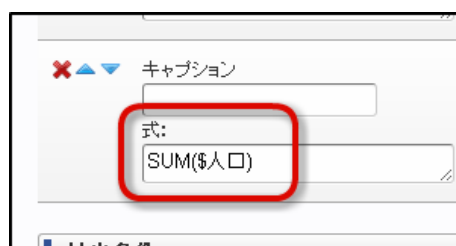
### ! 注意

式の実行結果を確認するには、クエリからデータが取得できる状態である必要があります。

### i コラム

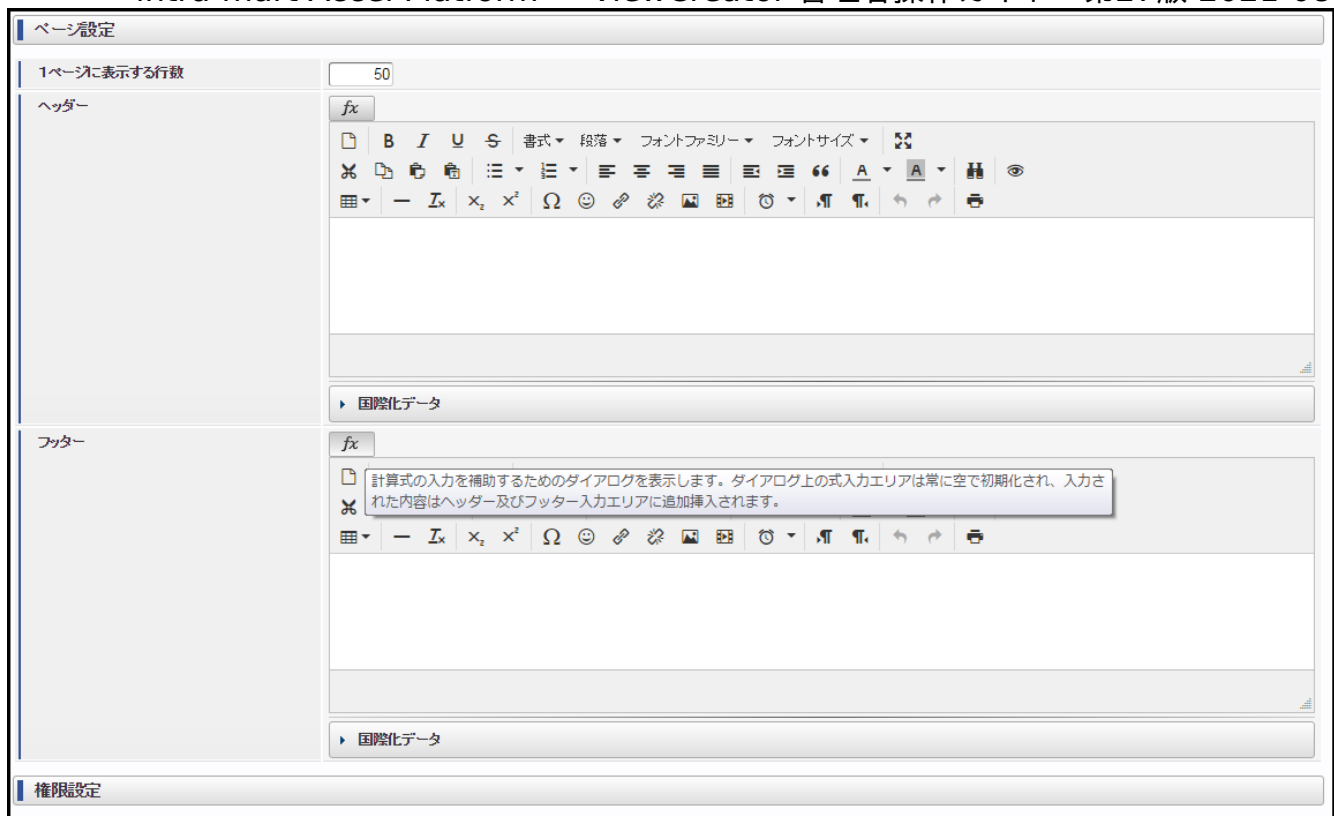
コラムの指定にはキャプションの代わりに内部的に生成されたコードを使用することもできます。キャプションを利用した場合、そのキャプションが変更されると式が実行できなくなってしまいますがコードを利用するとキャプションの変更に影響されることなく式の実行が可能です。但し式の可読性が落ちるため、適宜使い分けてご利用ください。

ダイアログ下部の「OK」ボタンをクリックすると、ダイアログが閉じて入力した式が設定されます。



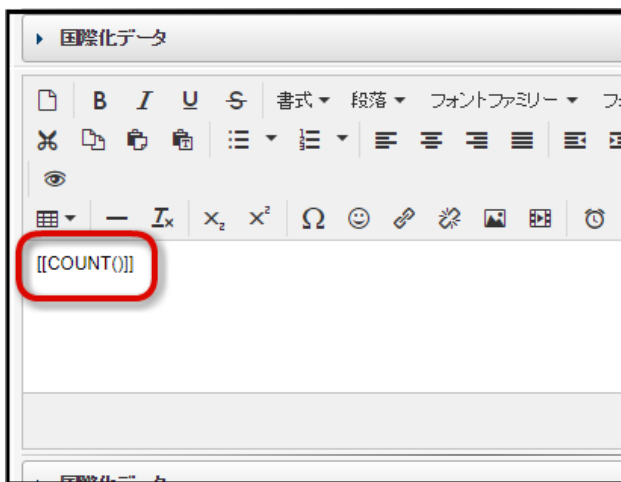
## ヘッダとフッタの計算式設定

計算式はヘッダとフッタに埋め込むことも可能です。



カラムの計算式と同様のダイアログが表示され、ダイアログを閉じると入力した計算式が設定されますが以下の点が異なります。

- ダイアログ上の式入力欄は常に空で初期化されます
- 入力された式は [[ と ]] で括られた状態で設定されます
- 常に追加挿入形式で設定されます



実行時には、[[ と ]] で括られた内部のみが式として解釈されて実行されます。

例)

設定例	実行結果例
レコード総数 : [[COUNT()]]件	レコード総数 : 110件



**注意**

ヘッダおよびフッタに設定する式の中では、レコード単位で取得するデータは使用できません。

ヘッダとフッタでは計算式に対して文字装飾を適用できます。

■ 設定例

国際化データ

fx

書式 段落 フォントファミリー フォントサイズ

レコード 総数	[[COUNT()]]件
日付	[[FORMAT('GGGGyyyy年MM月dd日', TODAY(), true, 'ja_JP_JP')]]

パス:

国際化データ

■ 表示例

地域 - 都道府県 - 年度 グループ化リスト(人口)

CSV出力 検索

1

地域名	都道府県名	年度	年齢	人口(合計)	人口密度+2	平均との差(人口)	
+ 近畿	-	-	-	149,646	★	34	148805
+ 関東	-	-	-	252,118	★	57	251277
+ 中部	-	-	-	141,850	★	23	141009
+ 北海道	-	-	-	38,752		2	37911
+ 四国	-	-	-	28,852	★	8	28011
+ 中国	-	-	-	52,922	★	10	52081
+ 九州	-	-	-	98,906	★	21	98065
+ 東北	-	-	-	66,938	★	8	66097
				829,984			

レコード総数	987件
日付	平成26年01月31日

! 注意

計算式に対して文字装飾を設定する場合は、計算式の中身だけでなく [[ と ]] も含めて纏めて同じ設定を適用してください。

※[[と]]の内部に文字装飾用のタグ文字列が混じると、演算ができない場合があります。

利用可能な関数一覧

ARRAY (定数値...)

i コラム

ARRAY関数は2014 Summer(Honoka)から利用可能です。

- 戻り値の型：配列型

引数を要素を持つ新しい配列を生成します。引数に配列が指定された場合は、その配列が持つ全ての要素が新しい配列に追加されます。

例：ARRAY(“あ”, “い”, “う”) -> “あ”, “い”, “う” の3つの文字列要素を持つ配列データが生成されます。

### AVG (\$カラム名)

- 戻り値の型：数値型

指定されたカラムの全レコードの平均値を返します。

### COLUMNVALUES (評価式, \$取得対象カラム, [最大取得データ数], [データ参照コード])



#### コラム

COLUMNVALUES関数は2014 Summer(Honoka)から利用可能です。



#### 注意

COLUMNVALUES関数はRDBのデータソースのみ使用可能です。

- 戻り値の型：配列型

データ参照から評価式に合致するレコードの取得対象カラムを配列で取得します。

データ参照コードが省略された場合はカレントのデータ参照から取得します。

※COLUMNVALUES関数はデータ参照のグループ化を解除して実行されるため、重複したデータが出力されることがあります。

例) COLUMNVALUES(\$面積 > 10000, \$都道府県名, 5, “xxxxxxxxxx”) -> “北海道”, “岩手”, “福島”, “長野”, “新潟”

例) COLUMNVALUES(\$地域名 == “関東” && \$人口 > 500, \$都道府県名) -> “東京”, “神奈川”, “埼玉”, “千葉”

### CONCATENATE (定数値...)

- 戻り値の型：文字列型

引数で指定された値を文字列連結します。

例) CONCATENATE(‘a’, ‘b’, ‘c’) -> abc

### COUNT ()

- 戻り値の型：数値型

レコード総数を返します。

### COUNTIF (評価式)

※この関数はRDBのデータソースのみ使用可能です

- 戻り値の型：数値型

評価式に合致するレコード数を返します。

例) COUNTIF(\$地域名 == “関東”) -> 7

### DATE (ミリ秒を表す数値orフォーマットパターン文字列, フォーマットパターン形式の文字列)

- 戻り値の型：日付型

日付型データを生成します ミリ秒を表す数値または、パターン文字列と文字列形式の日付データを引数として受け取ります。

例) DATE (1427846400000) -> 2015年04月01日 を表す日付型データ

例) DATE (“yyyy/MM/dd”, “2015/12/01”) -> 2015年12月01日 を表す日付型データ

**DATETIMEDURATIONFORMAT** (パターン文字列, ミリ秒の時間データ)

- 戻り値の型: 文字列型

時間データ (ミリ秒) をパターン文字列の形式でフォーマットします。

パターン文字列に使用可能な特殊文字は d (日) H (時間) m (分) s (秒) S (ミリ秒) の5つです。

例) DATETIMEDURATIONFORMAT('d日', 172800000) -> 2日

**EVAL** (評価式)

- 戻り値の型: 真偽値型

評価式の結果をTRUE/FALSEで返します。

**FORMAT** (フォーマット文字列, 数値または日付データ, [数値データを日付データのミリ秒として扱うかどうか true/false], [ロケールを表す文字列])

- 戻り値の型: 文字列型

数値または日付データを指定されたフォーマットで文字列に変換します。

ロケールを表す文字列は、言語コード、国コード、拡張情報をアンダースコアで連結した文字列を指定します。

この設定は、javaのAPI (java.util.Localeクラスのコンストラクタ) を通して変換処理に適用されます。

具体的には、"ja\_JP\_JP"を設定することで和暦形式の表示が可能です。

※この変換処理の動作仕様はjavaの実装に依存します。

例) FORMAT('yyyy/MM/dd', \$RECORD\_DATE) -> 2013/07/01

例) FORMAT('#,###', 123456789) -> 123,456,789

例) FORMAT('yyyy/MM/dd', 1369000000000, true) -> 2013/05/20

例) FORMAT('GGGGyyyy年', 1369000000000, true, "ja") -> 西暦2013年

例) FORMAT('GGGGyyyy年', 1369000000000, true) -> 西暦2013年

例) FORMAT('GGGGyyyy年', 1369000000000, true, "ja\_JP\_JP") -> 平成25年

**コラム**

[ロケールを表す文字列]の設定は2014 Spring(Granada)から利用可能です。

**IF** (評価式, 戻り値1, 戻り値2)

- 戻り値の型: 引数として渡された、戻り値1または戻り値2のデータ型

評価式がTRUEの場合は戻り値1、それ以外の場合は戻り値2を返します。

例) IF(\$都道府県名 == "東京", 1, 2)

都道府県名が東京である場合は1、それ以外の場合は2が返却されます。

**LEN** (文字列型の定数値)

- 戻り値の型: 数値型

指定された値の文字列長を返します。

**LOOKUP** (評価式, \$取得対象カラム名, [データ参照コード])

※この関数はRDBのデータソースのみ使用可能です。

- 戻り値の型: 取得対象カラムのデータ型

データ参照から評価式に合致するレコードの取得対象カラムを取得します。

データ参照コードが省略された場合はカレントのデータ参照から取得します。

例) LOOKUP(\$面積 > 10000, \$都道府県名, "xxxxxxxxxx") -> 北海道

**MAX** (\$カラム名)

- 戻り値の型：数値型

指定されたカラムの全レコード中の最大値を返します。

### MIN (\$カラム名)

- 戻り値の型：数値型

指定されたカラムの全レコード中の最小値を返します。

### NUM (定数値)

- 戻り値の型：数値型

指定された値を数値型に変換します。

日付型データが指定された場合はミリ秒データを取得します。

例：NUM("123")

実行結果例：123

例：NUM(TODAY())

実行結果例：1427400399551

### ROWNUMBER ()



#### コラム

ROWNUMBER関数は2019 Winter(Xanadu)から利用可能です。



#### 注意

ROWNUMBER関数をカラム一覧に複数設定したり、ヘッダおよびフッタに設定した場合は正しい行番号を表示できません。

- 戻り値の型：数値型

行番号表示用の数値を返します。

前ページまでのレコード数を初期値として呼び出されるたびに1ずつ加算されます。

### STR (定数値)

- 戻り値の型：文字列型

指定された値を文字列に変換します。

### SECUREPARAMETER (キー名, 値)



#### コラム

SECUREPARAMETER関数は2015 Spring(Juno)から利用可能です。

- 戻り値の型：文字列型

<imart type="secureParameter">タグで生成可能な暗号化されたデータを生成します。

キー名：secureParameterタグのname属性に相当するデータ

値：secureParameterタグのvalue属性に相当するデータ

### SUBSTITUTE (文字列, 検索文字列, 置換文字列)

- 戻り値の型：文字列型

第1引数で指定された文字列の中から検索文字列で指定された値を置換文字列で置換した結果を返します。

検索文字列には正規表現が使用できます。

例) SUBSTITUTE(" abc ", "s", "") -> "abc"

例) SUBSTITUTE("abc123", "[a-z]", "0") -> "000123"

**SUBSTRING (対象文字列, 開始インデックス番号, [終了インデックス番号])**

- 戻り値の型: 文字列型

部分文字列を取得します。

終了インデックス番号が省略された場合は対象文字列の長さが設定されます。

**SUBTRACT (引数1, 引数2)**

- 戻り値の型: 数値型

引数1から引数2を差し引いた数値データを取得します。

日付型データが渡された場合はミリ秒の数値として扱われます。

**SUM (\$カラム名) または SUM(定数値...)**

- 戻り値の型: 数値型

指定されたカラムの全レコードの合計を返します。

定数値が渡された場合は、それらの合計を返します。

例: SUM (\$人口)

実行結果例: 829984

例:SUM (1, 2, 3) -> 6

**SUMIF (評価式, \$合計値を計算するカラム名)**

※この関数はRDBのデータソースのみ使用可能です。

- 戻り値の型: 数値型

評価式に合致するレコードのみの合計値を取得します。

例: SUMIF (面積 > 7000, \$人口)

実行結果例: 248003

**SWITCH (判定対象値, 定数値1, 返却値1, [定数値2], [返却値2] ... [デフォルト値])****コラム**

SWITCH関数は2014 Summer(Honoka)から利用可能です。

- 戻り値の型: 引数として渡された、返却値、デフォルト値のデータ型

判定対象値を定数値と比較して、一致した場合に対応する返却値を返します。

例: SWITCH (都道府県名, "東京", 1, "神奈川", 2, "埼玉", 3, 4)

都道府県名 が 東京 である場合は、1 神奈川である場合は、2 埼玉である場合は、3、それ以外の場合は、4が返却されます。

**TIMEZONE ([取得対象], [取得情報])****コラム**

TIMEZONE関数は2020 Summer(Zephirine)から利用可能です。

- 戻り値の型: 引数で指定されたタイムゾーン情報

引数で指定されたタイムゾーン情報を返します

第1引数には、取得する対象を指定します。以下のいずれかのみ指定可能です。引数を省略した場合は"CSV"として扱われます。

"CSV": CSV出力のタイムゾーン

"DSP": 画面表示のタイムゾーン

第2引数には、取得する情報を指定します。以下のいずれかのみ指定可能です。引数を省略した場合は"GMT"として扱われます。

"GMT": GMT時差 ("GMT" + 時差 (+-HH:mm))

"ID": タイムゾーンID



**TODAY** ([タイムゾーンを表す文字列])

- 戻り値の型：日付型

カレントユーザに基づくシステム日付を取得します ※引数にタイムゾーンを表す文字列を指定することも可能です。

**VCSYSTEMPARAMETER** (動的パラメータを表すタグ文字列)
 コラム

VCSYSTEMPARAMETER関数は2014 Summer(Honoka)から利用可能です。

- 戻り値の型：文字列型

動的パラメータを表すタグ文字列を解析して結果データを文字列で取得します。

例： VCSYSTEMPARAMETER("<%USER\_ID%>")

実行結果例： aoyagi

例： VCSYSTEMPARAMETER("<%LOCALE%>")

実行結果例： ja

例： VCSYSTEMPARAMETER("<%ENCODING%>")

実行結果例： UTF-8

その他の引数に指定可能なタグ文字列については、「[動的パラメータ](#)」を参照してください。

 コラム

評価式とは、式全体でtrueまたはfalseと判定される計算式のことです。

例) \$SORT\_KEY > 10

**ユーザ定義関数**

関数は独自に定義して作成できます。

※新しい関数の作成にはJavaコードの記述が必要です。

新しい関数1つに対してJavaクラスを1つ作成します。

関数として登録するクラスは次のインタフェースを実装します

- jp.co.intra\_mart.foundation.viewcreator.formula.FunctionImplementation

このインタフェースには3つのメソッドが定義されています。

- void init(ViewCreatorData data);

ViewCreatorの設定データが渡されてきます。

ViewCreatorDataの中には、クエリの設定、データ参照の設定、データ参照表示時にユーザが入力した検索条件やソート列の情報が含まれます。

- String getName()

関数名を取得します。

※他の関数と名称が重複しないように注意してください。

- FixedValue execute(Arguments args) throws FunctionExecuteException;

関数が実行されたとき呼び出されます。

渡された引数に対して戻り値を返します。

実行エラーが発生した場合は例外をスローします。



## コラム

引数や戻り値として使用可能な型については

jp.co.intra\_mart.foundation.viewcreator.model.formulaパッケージ内のクラスを参照してください

- 関数の登録  
作成したクラスの情報を下記の設定ファイルに登録します。

%コンテキストパス%/WEB-INF/conf/viewcreator-function-config.xml

```
<function-config>
  <class-name>%作成したクラス名%</class-name>
</function-config>
```

設定を反映するにはサーバの再起動が必要です。

- 入力ダイアログ上で引数や内容の説明を記載する

メッセージプロパティファイルに以下のキーを使用して文章を作成してください

引数の説明 : MSG.I.IWP.VIEWCREATOR.DATAVIEWEDIT.FUNCTIONS.%関数名%.ARGUMENT

内容の説明 : MSG.I.IWP.VIEWCREATOR.DATAVIEWEDIT.FUNCTIONS.%関数名%.DESCRIPTION

## リスト集計におけるCSV出力の設定

リスト集計のCSV出力機能は設定ファイルを編集することで、出力されるファイルの文字コードや改行コードなどを変更できます。

### 目次

- 設定ファイルの配置場所
- 設定項目
  - フォーマット適用の優先順について
- viewcreator-config.xmlで設定可能な項目について
  - 設定項目

### 設定ファイルの配置場所

%PUBLIC\_STORAGE\_PATH%/products/viewcreator/list-dataview-export-option.xml

### 設定項目

基本的にはTableMaintenanceのテーブル・エクスポート機能と同様の設定が可能です。

設定項目の詳細については「[テーブルメンテナンス管理者ガイド - テーブル・エクスポート - オプション設定について](#)」を参照してください。

```

<export-option>
  <!-- DATE型、TIMESTAMP型データをテキストデータに変換するためのフォーマット -->
  <date-format>yyyy-MM-dd HH:mm:ss</date-format>
  <!-- 数値系データをテキストデータに変換するためのフォーマット -->
  <decimal-format>#</decimal-format>
  <!-- エクスポートファイルで利用する改行コード CR, LF, CRLFのいずれか -->
  <newline-code>CRLF</newline-code>
  <!-- エクスポートファイルのエンコーディング -->
  <encoding>MS932</encoding>
  <!-- バイナリ系のデータを別ファイルへエクスポートするかどうか -->
  <include-binary>>false</include-binary>
  <!-- エクスポートしたファイルを常にzipで圧縮するかどうか -->
  <zip-file>>true</zip-file>
  <!-- エクスポート時の最大出力レコード数 -->
  <max-row>0</max-row>
  <!-- エクスポート時にStatement, ResultSetに設定するフェッチサイズ -->
  <fetch-size>10</fetch-size>
  <!-- null値に対する置換文字列 -->
  <null-value/>
  <!-- 未使用 -->
  <output-footer>>false</output-footer>
  <!-- ヘッダを出力するかどうか -->
  <output-header>>true</output-header>
  <!-- 未使用 -->
  <sleep-record-count>0</sleep-record-count>
  <!-- 未使用 -->
  <sleep-time>0</sleep-time>
  <!-- 未使用 -->
  <start-index>0</start-index>
  <!-- zip圧縮時のファイル名 -->
  <zip-file-name/>
  <table>
    <table-name>vcexport_sql</table-name>
    <!-- 出力ファイル（CSVファイル）のファイル名。 [[ と ]] で囲われた文字列は計算式として解釈されます。 -->
    <dest-file-name>[[DATAVIEWCODE()]]_[[FORMAT("yyyyMMddHHmmss", TODAY())]]</dest-file-name>
  </table>
  <exporter>
    <exporter-name>vcexporter</exporter-name>
    <!--
      出力したデータをダブルコーテーションで囲うかどうか
      true: 文字列型のデータを常にダブルコーテーションで囲います
      false: データの中にエスケープが必要な文字が含まれる場合のみダブルコーテーションで囲います
    -->
    <param>
      <param-name>useDoubleQuotation4AllStringValue</param-name>
      <param-value>>false</param-value>
    </param>
  </exporter>
</export-option>

```

date-format	日時データのデフォルトの出力フォーマットを設定します。 データ参照編集のカラム設定でフォーマット文字列が指定されていない場合のみ適用されます。 この設定はタイムスタンプ型または日付型のカラムに対してのみ有効です。 ※画面上のフォーマット設定項目で指定された値が優先されます。
decimal-format	数値データのデフォルトの出力フォーマットを設定します。 データ参照編集のカラム設定でフォーマット文字列が指定されていない場合のみ適用されます。 この設定は数値型のカラムに対してのみ有効です。 ※画面上のフォーマット設定項目で指定された値が優先されます。
newline-code	レコード単位の改行文字を設定します。 CR, LF, CRLFのいずれかを設定します。
encoding	出力ファイルの文字エンコーディングセットを設定します。
include-binary	バイナリ系のデータを別ファイルへエクスポートするかを設定します。

zip-file	エクスポートしたファイルを常にzipで圧縮するかどうか。 ※画面上のダイアログで指定された値が優先されます。 conf/viewcreator-config.xmlで、export-data-settings/show-confirm-zipの設定値がfalseのときに適用されます。
max-row	出力する最大レコード数を設定します。0を設定した場合は、最大レコード数は無制限として扱われます。
fetch-size	StatementやResultSetに設定するフェッチサイズです。フェッチサイズの値を変更することによりパフォーマンスが向上することがあります。 レコード数が多い場合、フェッチサイズの値を何度か変更して実行し、最適になる値を決定してください。
null-value	データがnull値のときに出力する文字列を設定します。
output-footer	未使用
output-header	カラムヘッダを出力するかどうかを設定します。
sleep-record-count	未使用
sleep-time	未使用
start-index	出力を開始するレコード位置を設定します。
zip-file-name	zip圧縮された場合のzipファイル名（拡張子なし）を設定します。 省略時はシステムで自動生成された値が利用されます。
table/table-name	vcexport_sqlで固定です。変更はできません。
table/dest-file-name	エクスポートファイルのファイル名（拡張子なし）を設定します。 [[ と ]] で囲われた文字列は計算式として解釈されます。
exporter/exporter-name	vcexporterで固定です。変更はできません。
param/param-name/useDoubleQuotation4AllStringValue	文字列データをダブルコーテーションで常に囲うかどうかを設定します。 true: 文字列型のデータを常にダブルコーテーションで囲います。 false: データの中にエスケープが必要な文字が含まれる場合のみダブルコーテーションで囲います。
use-bom	BOMを付与するかどうかを設定します。 BOMは出力ファイルの文字エンコーディングセットがUTF-8の場合のみ利用可能です。 true: 出力ファイルにBOMを付与します。 false: 出力ファイルにBOMを付与しません。

### コラム

dest-file-nameは2014 Winter(Iceberg)から利用可能です。  
useDoubleQuotation4AllStringValueは2014 Winter(Iceberg)から利用可能です。  
fetch-sizeは2017 Winter(Rebecca)から利用可能です。

### 注意

use-bomは2016 Spring(Maxima)から利用可能です。  
標準では設定ファイルに設定されておりませんので、必要に応じて設定ファイルを編集しご利用ください。

## フォーマット適用の優先順について

数値型、日付型、タイムスタンプ型に対して適用されるフォーマットの優先順は以下の通りです。

- 画面表示の場合
  1. データ参照編集画面で設定されているフォーマット
  2. 個人設定の日付と時刻の形式で設定されているパターン（日付型とタイムスタンプ型）
- 「CSV出力」の場合
  1. データ参照編集画面で設定されているフォーマット

## viewcreator-config.xmlで設定可能な項目について

CSV出力に関する一部の設定はviewcreator-config.xmlで行います。

ファイルパス：

WEB-INF/conf/viewcreator-config.xml

## 設定項目

export-data-settings/show-confirm-zip

出力するときにzip圧縮選択のダイアログを表示するかどうかを設定します。

true:表示する

false:表示しない



コラム

show-confirm-zipは2014 Winter(Iceberg)から利用可能です。

## CSV出カジョブの登録

リスト集計および、サマリ集計のデータ参照からCSVファイルを出力するジョブを登録します。



コラム

CSV出カジョブの登録は2014 Summer(Honoka)から利用可能です。

## 目次

- 設定可能なパラメータ

データ参照一覧から、ジョブ登録するリスト集計またはサマリ集計のデータ参照を選択し、「CSV出カジョブの登録」を選択します。

<input type="checkbox"/>	編集	データ参照	データ参照名	更新日	作成者
<input type="checkbox"/>			2000年度版・地域 - 都道府県 サマリ	2013/11/05 22:18	tenant
<input type="checkbox"/>			2000年度版・地域 - 都道府県 サマリ	2013/10/31 18:20	tenant
<input type="checkbox"/>			[ポータル用]トランザクションログ・ページ	2014/06/13 01:16	tenant
<input type="checkbox"/>			エラーログ	2014/06/13 01:16	tenant
<input type="checkbox"/>			サマリ集計から作成した棒グラフサン	2013/11/06 16:29	tenant
<input type="checkbox"/>			システムログ	2014/06/13 01:16	tenant
<input type="checkbox"/>			データ参照一覧・サンプル1	2013/11/06 18:35	tenant

サイトマップから「ジョブ管理 - ジョブ設定」を選択します。

ジョブとジョブカテゴリ(ViewCreator)が登録されていることが確認できます。

## 設定可能なパラメータ

dataview_cd	CSVを出力する対象のデータ参照コードです。
dest_storage_path	出力先のパブリックストレージパスです。
dest_storage_file_name	出力ファイル名（拡張子なし）です。[[ と ]] で囲われた文字列は計算式として解釈されます。未指定の場合、出力するファイル名は「dataview_cd」 + “_” + 「現在日時（yyyyMMddHHmmssSSS形式）」 + 「.csv」または“.zip”です。
zip	出力ファイルをZIPで圧縮するかどうかを設定します。 true: 圧縮する false: 圧縮しない

### コラム

#### dest\_storage\_path について

- 2017 Winter(Rebecca) (ViewCreator 8.0.17) までは、dest\_storage\_path に dataview\_cd と日時を連結したものをファイル名として出力します。  
 例1: dest\_storage\_path 「vc\_list」、dataview\_cd 「mylist」、日付「2018年1月23日9時20分12.345秒」で zip 出力の場合、出力先はパブリックストレージのルートで、ファイル名は「vc\_listmylist\_20180123092012345.zip」です。  
 例2: dest\_storage\_path 「vc\_list/」、dataview\_cd 「mylist」、日付「2018年1月23日9時20分12.345秒」で zip 出力の場合、出力先はパブリックストレージの「vc\_list」で、ファイル名は「mylist\_20180123092012345.zip」です。
- 2017 Winter(Rebecca) (ViewCreator 8.0.17-PATCH\_001) からは、dest\_storage\_path に指定したパブリックストレージのディレクトリに出力します。  
 例: dest\_storage\_path 「vc\_list/list01」、dataview\_cd 「mylist」、日付「2018年1月23日9時20分12.345秒」で zip 出力の場合、出力先はパブリックストレージの「vc\_list/list01」で、ファイル名は「mylist\_20180123092012345.zip」です。

**i** コラム

dest\_storage\_file\_name は、2018 Spring(Skylark) から利用可能です。

**!** 注意

ジョブネットとジョブネットカテゴリは自動生成されません。  
適宜作成してください。

## ソースコード出力

## ■ ソースコード出力とは？

作成済みのデータ参照の設定データを元に個別で実行可能なスクリプト開発のソースコードとして生成・出力する機能です。  
出力されたソースコードは、適宜カスタマイズして利用できます。  
また、カスタマイズは出力元のデータ参照に影響を及ぼすことなく行うことができます。

**i** コラム

ソースコード出力は2014 Summer(Honoka)から利用可能です。

## 目次

- ソースコードを出力するための手順
- 生成されたソースコードを動作させるための手順
  - ソースコードの移動とルーティングの設定 (2015 Winter(Lydia)以前の場合)
  - ソースコードの移動・修正とルーティングの設定 (2016 Spring(Maxima)から2017 Winter(Rebecca)までの場合)
    - jsファイルの編集について
    - ルーティングの設定について
  - ソースコードの移動・修正とルーティングの設定 (2018 Spring(Skylark)以降の場合)
    - jsファイルの編集について
    - ルーティングの設定について

## ソースコードを出力するための手順

ソースコード出力を行う対象とするデータ参照を選択します。

操作は「データ参照一覧」画面で行います。

- データ参照一覧から、データ参照のチェックボックスを選び「ソースコード出力」を選択します。

<input type="checkbox"/>	編集	データ参照	データ参照名	更新日	作成者
<input type="checkbox"/>			2000年度版・地域 - 都道府県 サマリ	2013/11/05 22:18	tenant
<input type="checkbox"/>			2000年度版・地域 - 都道府県 サマリ	2013/10/31 18:20	tenant
<input type="checkbox"/>			[ポータル用]トランザクションログ・ページ	2014/06/13 01:16	tenant
<input type="checkbox"/>			エラーログ	2014/06/13 01:16	tenant
<input type="checkbox"/>			サマリ集計から作成した棒グラフサン	2013/11/06 16:29	tenant
<input type="checkbox"/>			システムログ	2014/06/13 01:16	tenant
<input type="checkbox"/>			データ参照一覧・サンプル1	2013/11/06 18:35	tenant
<input type="checkbox"/>			データ参照一覧・サンプル2(クエリ別	2013/11/06 18:35	tenant

以下の4つのファイルがパブリックストレージに出力されます。

1. クエリの設定データ (XMLファイル)
2. データ参照の設定データ (XMLファイル)
3. スクリプト開発のソースコード (jsファイルとhtmlファイル)

## 生成されたソースコードを動作させるための手順

生成されたソースコードを実際に動作させる手順を示します。  
パブリックストレージに出力されたファイルのうち、jsファイルとhtmlファイルの再配置やルーティングの設定が必要です。



### 注意

2018 Spring(Skylark)以降と、2016 Spring(Maxima)から2017 Winter(Rebecca)までと、2015 Winter(Lydia)以前で設定手順が異なります。

### ソースコードの移動とルーティングの設定 (2015 Winter(Lydia)以前の場合)

1. jsファイルとhtmlファイルをWEB-INF/jsspディレクトリ以下の任意のディレクトリに移動する
2. 移動先のパスに対して、スクリプト開発モデルルーティングの設定を行う

ルーティングについての詳細は、[スクリプト開発モデルルーティング設定](#) を参照してください。

### ソースコードの移動・修正とルーティングの設定 (2016 Spring(Maxima)から2017 Winter(Rebecca)までの場合)

1. jsファイルとhtmlファイルをWEB-INF/jsspディレクトリ以下の任意のディレクトリに移動する
2. jsファイルを編集する
3. 移動先のパスに対して、スクリプト開発モデルルーティングの設定を行う

### jsファイルの編集について

出力されたファイルのうち、jsファイルについては下記の変数の値を修正してください。

※設定した値を、ルーティングの設定と合わせる必要があります。

1. SELF\_JSSP\_PATH  
スクリプト開発モデルプログラムのソースディレクトリからの相対パス形式で、移動先のファイルパスをセットします。  
拡張子の指定はしないでください。
2. OUTPUT\_REPORT\_URL  
帳票出力を実行するためのURLをセットします。
3. OUTPUT\_CSV\_URL  
CSV出力を実行するためのURLをセットします。  
※クロス集計とグラフ集計の場合は不要です。

#### ■ 編集前jsファイルの例

```
// 自分自身のJSSPファイルパス (*拡張子無し)
var SELF_JSSP_PATH = "sample/gen_5i7urzmesvqqip4";

// 帳票出力用URL
var OUTPUT_REPORT_URL = "sample/5i7urzmesvqqip4/pdf";

// CSV出力用URL
var OUTPUT_CSV_URL = "sample/5i7urzmesvqqip4/csv";
```

### ルーティングの設定について

下記のルーティングを作成してください。

1. 参照画面の表示用のマッピング  
path属性には、データ参照の表示画面にアクセスするためのURLをセットします。  
page属性には、jsファイルのSELF\_JSSP\_PATHと同じ値をセットします。



## 2. 帳票出力用のマッピング

path属性には、jsファイルのOUTPUT\_REPORT\_URLと同じ値をセットします。

page属性には、jsファイルのSELF\_JSSP\_PATHと同じ値をセットします。

action属性には、“outputReport”をセットします。

## 3. CSV出力用のマッピング ※クロス集計とグラフ集計の場合は不要です。

path属性には、jsファイルのOUTPUT\_CSV\_URLと同じ値をセットします。

page属性には、jsファイルのSELF\_JSSP\_PATHと同じ値をセットします。

action属性には、“outputCSV”をセットします。

## ■ 設定例

```
<!-- 参照画面の表示用のマッピング -->
<file-mapping path="%参照画面を表示する任意のURL%" page="%SELF_JSSP_PATHの設定値%">
...
</file-mapping>
<!-- 帳票出力用のマッピング action属性を、"outputReport"としてください-->
<file-mapping path="%OUTPUT_REPORT_URLの設定値%" page="%SELF_JSSP_PATHの設定値%" action="outputReport">
...
</file-mapping>
<!-- CSV出力用のマッピング action属性を、"outputCSV"としてください -->
<file-mapping path="%OUTPUT_CSV_URLの設定値%" page="%SELF_JSSP_PATHの設定値%" action="outputCSV">
...
</file-mapping>
```

ルーティングについての詳細は、[スクリプト開発モデルルーティング設定](#) を参照してください。

## ソースコードの移動・修正とルーティングの設定（2018 Spring(Skylark)以降の場合）

1. jsファイルとhtmlファイルをWEB-INF/jsspディレクトリ以下の任意のディレクトリに移動する
2. jsファイルを編集する
3. 移動先のパスに対して、スクリプト開発モデルルーティングの設定を行う

## jsファイルの編集について

出力されたファイルのうち、jsファイルについては下記の変数の値を修正してください。

※設定した値を、ルーティングの設定と合わせる必要があります。

## 1. SELF\_JSSP\_PATH

スクリプト開発モデルプログラムのソースディレクトリからの相対パス形式で、移動先のファイルパスをセットします。  
拡張子の指定はしないでください。

## 2. VIEW\_PAGE\_URL

参照画面（自分自身）を表示するURLをセットします。

## 3. OUTPUT\_REPORT\_URL

帳票出力を実行するためのURLをセットします。

## 4. OUTPUT\_CSV\_URL

CSV出力を実行するためのURLをセットします。

※クロス集計とグラフ集計の場合は不要です。

## ■ 編集前jsファイルの例

```
// 自分自身のJSSPファイルパス（※拡張子無し）
var SELF_JSSP_PATH = "sample/gen_5i7urzmesvqqip4";

// 参照画面（自分自身）用URL
var VIEW_PAGE_URL = "sample/5i7urzmesvqqip4";

// 帳票出力用URL
var OUTPUT_REPORT_URL = "sample/5i7urzmesvqqip4/pdf";

// CSV出力用URL
var OUTPUT_CSV_URL = "sample/5i7urzmesvqqip4/csv";
```

## ルーティングの設定について

下記のルーティングを作成してください。

1. 参照画面の表示用のマッピング  
path属性には、jsファイルのVIEW\_PAGE\_URLと同じ値をセットします。  
page属性には、jsファイルのSELF\_JSSP\_PATHと同じ値をセットします。
2. 帳票出力用のマッピング  
path属性には、jsファイルのOUTPUT\_REPORT\_URLと同じ値をセットします。  
page属性には、jsファイルのSELF\_JSSP\_PATHと同じ値をセットします。  
action属性には、“outputReport”をセットします。
3. CSV出力用のマッピング ※クロス集計とグラフ集計の場合は不要です。  
path属性には、jsファイルのOUTPUT\_CSV\_URLと同じ値をセットします。  
page属性には、jsファイルのSELF\_JSSP\_PATHと同じ値をセットします。  
action属性には、“outputCSV”をセットします。

- 設定例

```
<!-- 参照画面の表示用のマッピング -->
<file-mapping path="%VIEW_PAGE_URLの設定値%" page="%SELF_JSSP_PATHの設定値%">
...
</file-mapping>
<!-- 帳票出力用のマッピング action属性を、"outputReport"としてください-->
<file-mapping path="%OUTPUT_REPORT_URLの設定値%" page="%SELF_JSSP_PATHの設定値%" action="outputReport">
...
</file-mapping>
<!-- CSV出力用のマッピング action属性を、"outputCSV"としてください -->
<file-mapping path="%OUTPUT_CSV_URLの設定値%" page="%SELF_JSSP_PATHの設定値%" action="outputCSV">
...
</file-mapping>
```

ルーティングについての詳細は、[スクリプト開発モデルルーティング設定](#) を参照してください。

## 帳票出力の設定

データ参照をPDF等の形式で出力することを可能にする設定です。  
すべてのタイプのデータ参照で利用できます。

### コラム

帳票出力の設定は2014 Summer(Honoka)から利用可能です。

#### 目次

- [設定ファイルの配置場所](#)
- [設定項目](#)
- [コマンドを利用する場合に指定するパラメータ](#)
- [帳票出力の実装としてwkhtmltopdfを利用する場合のインストール手順と設定](#)

設定が有効の場合、データ参照表示時に「帳票出力」が表示されます。

		1990			1995			2000			合計	平均
		0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上	0 - 14歳	15 - 64歳	65歳以上		
北海道	北海道	1,034	3,925	675	899	3,943	845	792	3,833	1,032	16,978	1,886.44
	小計										16,978	1,886.44
東北	青森	289	1,001	192	252	991	237	223	965	287	4,437	493
	岩手	270	941	206	239	925	255	212	899	304	4,251	472.33
	宮城	439	1,535	267	394	1,596	338	354	1,602	409	6,934	770.44
	秋田	220	816	192	189	787	238	163	746	280	3,631	403.44
	山形	234	819	205	209	799	249	186	772	286	3,759	417.67
	福島	422	1,378	302	382	1,380	372	341	1,354	432	6,363	707
	小計											29,375

## 設定ファイルの配置場所

%コンテキストパス%/WEB-INF/conf/viewcreator-config.xml

## 設定項目

### 設定例

```
<report-settings>
  <report-creator-class-name></report-creator-class-name>
  <report-create-command>wkhtmltopdf %INPUT_PATH% %OUTPUT_PATH%</report-create-command>
  <file-suffix>pdf</file-suffix>
</report-settings>
```

### コラム

帳票出力の設定はデフォルトではコメントアウト（無効化）されています。

report-creator-class-name	HTMLを帳票ファイルへの変換処理を行うための、下記のインタフェースの実装クラス名を設定します。  jp.co.intra_mart.foundation.viewcreator.report.ReportFileConvertor  report-create-commandとreport-creator-class-nameはどちらか1つだけ設定してください
report-create-command	HTMLを帳票ファイルへの変換処理を行うための、実行可能なコマンドを設定します。  report-create-commandとreport-creator-class-nameはどちらか1つだけ設定してください
file-suffix	帳票ファイルの拡張子を設定します。

## コマンドを利用する場合に指定するパラメータ

%INPUT\_PATH% 帳票ファイルの変換元となるHTMLファイルのパス

%OUTPUT\_PATH% 帳票ファイルの出力先ファイルパス

これらのパラメータには、コマンド実行時に適切な値が自動的にセットされます。

**!** 注意

帳票ファイルの変換元となるHTMLファイルはアプリケーションサーバのローカル上に一旦作成されその後、変換処理を行うプログラム（クラスやコマンド）に渡されます。  
HTMLファイルから参照される静的コンテンツ（cssファイルや画像ファイル等）は、ベースURLを元にWebサーバから取得されます。  
そのため、アプリケーションサーバからWebサーバで公開されているコンテンツを参照できるように設定されている必要があります。

## 帳票出力の実装としてwkhtmltopdfを利用する場合のインストール手順と設定

wkhtmltopdfのインストール手順については、下記のマニュアルを参照してください。

「[wkhtmltopdfのインストール](#)」

<report-create-command>タグは、以下のように設定してください。

```
<report-create-command>wkhtmltopdf [オプション設定] %INPUT_PATH% %OUTPUT_PATH%</report-create-command>
```

**i** コラム

wkhtmltopdfで利用可能なオプションについては、helpコマンドで確認してください。

```
$ wkhtmltopdf -h
```

## null値に対する表示文字列の設定

null値をリスト集計等で表示する場合に、表示用の代替文字列を設定できます。

**i** コラム

null値に対する表示文字列の設定は2014 Summer(Honoka)から利用可能です。

## 目次

- [設定ファイルの配置場所](#)
- [設定項目](#)

## 設定ファイルの配置場所

```
%コンテキストパス%/WEB-INF/conf/viewcreator-config.xml
```

## 設定項目

設定例

```
<replacement-null-value> - </replacement-null-value>
```

円グラフ以外のグラフ集計（棒グラフや折れ線グラフ、レーダーチャート）では、凡例の表示方法を変更できます。

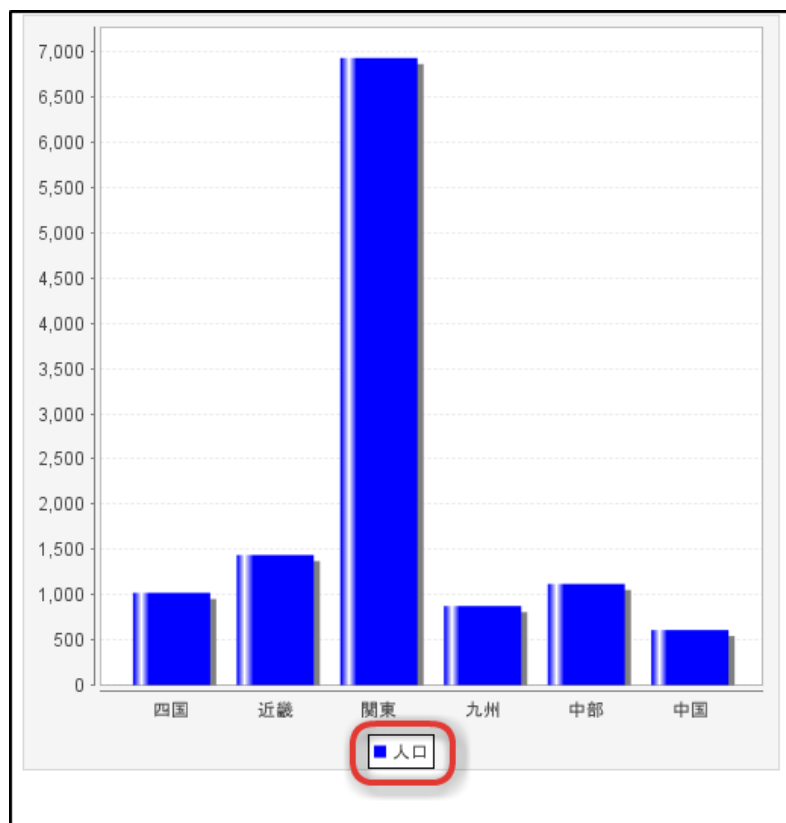
## i コラム

凡例の表示方法変更は2014 Winter(Iceberg)から利用可能です。

### 目次

- 凡例カラムの設定をしない場合
- 凡例カラムの設定を行った場合

## 凡例カラムの設定をしない場合



	人口
四国	1022.0
近畿	8789.0
関東	12017.0
九州	1309.0
関東	8475.0
九州	876.0
中部	1121.0
中国	613.0
近畿	1440.0
関東	6925.0

凡例として「人口」というデータが表示されています。

**カラム設定**

カラムの国際化項目の編集

キャプションカラム 地域名(name) ▼

凡例ロードカラム ▼

凡例ラベルカラム ▼

値カラム 年度(year) ▼

	表示	カラム	カラー
▲ ▼	<input type="checkbox"/>	年度	#000000
▲ ▼	<input type="checkbox"/>	面積	#ff0000
▲ ▼	<input checked="" type="checkbox"/>	人口	#0000ff
▲ ▼	<input type="checkbox"/>	人口密度	#000000

特に凡例についての設定をしない場合は、表示対象のカラムのカラム名が凡例の項目として採用されます。  
 ※8.0.8 - 2014 Summer (Honoka)以前についても同様の動きです。

#### 凡例カラムの設定を行った場合

**カラム設定**

カラムの国際化項目の編集

キャプションカラム 地域名(name) ▼

凡例ロードカラム 都道府県名(name) ▼

凡例ラベルカラム 都道府県名(name) ▼

値カラム 年度(year) ▼

	表示	カラム	カラー
▲ ▼	<input type="checkbox"/>	年度	#000000
▲ ▼	<input type="checkbox"/>	面積	#ff0000
▲ ▼	<input checked="" type="checkbox"/>	人口	#0000ff
▲ ▼	<input type="checkbox"/>	人口密度	#000000

凡例コードカラムで選択したカラムに格納されているデータ（上記の例では「都道府県名」）が凡例として利用されます。



**i** コラム

「凡例コードカラム」に設定されたカラムから取得されたデータをキーとして、同じ凡例項目に属するデータであるかどうかの判定が行われます。

画面に表示する値を別にしたい場合は、「凡例ラベルカラム」を設定してください。

## クエリ編集画面でテーブル単位のアクセス権を設定する

クエリ編集画面のテーブル一覧に表示されるテーブルにはアクセス権（参照権）を設定できます。

**i** コラム

テーブル単位のアクセス権設定は2014 Winter(Iceberg)から利用可能です。

**!** 注意

2020 Spring(Yorkshire)から追加されたSQLビルダはテーブル単位のアクセス権の制御はできません。SQLビルダでは接続しているデータベースのすべてのテーブルを利用できます。

クエリ編集

一時保存 保存 削除 SQLの表示 SQLで編集 Viewの作成 プレビュー 100%

テーブル一覧 ビュー一覧 リソース一覧

テーブル一覧

- b\_m\_account\_attr\_b
- b\_m\_account\_b
- b\_m\_account\_dtf
- b\_m\_account\_role\_b
- b\_m\_account\_theme
- b\_m\_calendar\_info\_b
- b\_m\_calendar\_info\_i
- b\_m\_calendar\_merge\_b
- b\_m\_calendar\_week\_b
- b\_m\_day\_info\_b
- b\_m\_day\_info\_i
- b\_m\_day\_info\_set\_b
- b\_m\_day\_info\_set\_i
- b\_m\_link\_category
- b\_m\_link\_info
- b\_m\_menu\_argument\_b
- b\_m\_menu\_b
- b\_m\_menu\_group\_b
- b\_m\_menu\_group\_cate\_inclusion

クエリ定義情報

クエリコード: 517ur3opljh2x  
クエリ名: ViewCreatorの設定テーブルを利用したクエリ  
説明

接続ID: default

Minimap

カラム一覧 抽出条件一覧

上へ	下へ	テーブルビュー/リソース	カラム/フィールド	型	ソート	キャプション	削除
↑	↓	b_vc_query	query_cd	文字列	指定なし	クエリコード	×
↑	↓	b_vc_query	query_name	文字列	指定なし	クエリ名	×
↑	↓	b_vc_data	data_cd	文字列	指定なし	データ参照コード	×
↑	↓	b_vc_data	data_name	文字列	指定なし	データ参照名	×
↑	↓	b_vc_data	record_user_cd	文字列	指定なし	更新者ユーザコード	×
↑	↓	b_vc_data	pattern	文字列	指定なし	データ参照パターン	×
↑	↓	b_vc_data_i	data_name	文字列	指定なし	データ参照名 (国際化)	×

目次

- 設定の手順

設定の手順

まず、アクセス権を設定するテーブルについて認可リソースとして登録を行います。  
認可リソースの登録は手作業で「認可設定」画面から実施することも可能ですが  
TableMaintenanceモジュールがインストールされている環境では、簡易な手順で登録を行うことができます。

「サイトマップ」→「TableMaintenance」→「テーブル一覧」を選択します。



テーブル一覧

認可設定画面を開く

tenant ▼  
b\_vc 検索 クリア

	テーブル名	説明	認可	仕様書
84.	B_VC_DATA			
85.	B_VC_DATA_ACL			
86.	B_VC_DATA_CROSS			
87.	B_VC_DATA_GRAPH			
88.	B_VC_DATA_GRAPH_DATA			
89.	B_VC_DATA_GRAPH_INFO			
90.	B_VC_DATA_I			
91.	B_VC_DATA_LIST_CONDITION			
92.	B_VC_DATA_LIST_SELECT			
93.	B_VC_DATA_LIST_SELECT_DISP			
94.	B_VC_DATA_LIST_SELECT_FORMULA			
95.	B_VC_DATA_LIST_SELECT_I			
96.	B_VC_QUERY			
97.	B_VC_QUERY_COLUMN			
98.	B_VC_QUERY_CONDITION			
99.	B_VC_QUERY_LINE			
100.	B_VC_QUERY_LINE_DETAIL			
101.	B_VC_QUERY_TABLE			

「認可」列のアイコンをクリックして認可リソースの登録を行います。  
登録が完了すると、アイコンが緑のチェックマーク表示に変わります。

テーブル一覧

認可設定画面を開く

tenant

b\_vc 検索 クリア

	テーブル名	説明	認可	仕様書
84.	B_VC_DATA		✓	
85.	B_VC_DATA_ACL		✓	
86.	B_VC_DATA_CROSS		✓	
87.	B_VC_DATA_GRAPH		✓	
88.	B_VC_DATA_GRAPH_DATA		✓	
89.	B_VC_DATA_GRAPH_INFO		✓	
90.	B_VC_DATA_I		✓	
91.	B_VC_DATA_LIST_CONDITION		✓	
92.	B_VC_DATA_LIST_SELECT		✓	
93.	B_VC_DATA_LIST_SELECT_DISP		✓	
94.	B_VC_DATA_LIST_SELECT_FORMULA		✓	
95.	B_VC_DATA_LIST_SELECT_I		✓	
96.	B_VC_QUERY		✓	

次に、登録された認可リソースに対して、アクセス権を設定します。

ツールバーから「認可設定画面を開く」を選択します。

テーブルのアクセス権設定

エクスポート キャッシュクリア

アクションの種類 全てのアクション

全て許可
  全て禁止
  全て未設定
 [条件の新規作成](#)

リソース	アクション	認証		ユーザ	組織		ロール			
		ゲストユーザ	認証済みユーザ	青柳辰巳	サンプル会社	その他会社	テナント管理者	認可管理者	メニュー管理者	メニュー運用管理者
画面・処理										
TableMaintenance										
テーブル一覧・一般ユーザ向け	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
テーブル単位の閲覧・編集	実行	✗	✗	✗	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_LIST_SELECT	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_LIST_SELECT_I	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_QUERY_CONDITION	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_QUERY_LINE	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_LIST_CONDITION	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_GRAPH_DATA	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_LIST_SELECT_FORMULA	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_LIST_SELECT_DISP	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_GRAPH_INFO	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗
tenant/B_VC_DATA_CROSS	実行	✗	✗	✓	✗	✗	✗	✗	✗	✗

テーブルへのアクセス権利リソースに対して、認可設定を行います。

これで設定完了です。

登録された認可リソースの詳細については「[TableMaintenance 管理者操作ガイド - 認可リソースの登録](#)」を参照してください。

## REST APIを参照する

ViewCreator のREST APIは intra-mart Accel Platform に組み込まれているSwagger UIで確認、実行が可能です。

Swagger UIは、以下のURLから確認できます。

`http(s)://{HOST}:{PORT}/{CONTEXT_PATH}/swagger_ui?url={CONTEXT_PATH}/api-docs/viewcreator`  
(プロトコル、ホスト名、ポート番号、コンテキストパスは環境にあわせて置き換えてください。)

例: `http://localhost:8080/imart/swagger_ui?url=/imart/api-docs/viewcreator`

「Category」をクリックするとルーティングカテゴリに関するAPIの一覧が表示されます。

「DataView」をクリックするとデータ参照に関するAPIの一覧が表示されます。

**DataView : データ参照に関するAPIです。** Show/Hide | List Operations | Expand Operations

DELETE	/api/viewcreator/views	データ参照を一括削除します。
GET	/api/viewcreator/views	データ参照定義の検索を行います。
POST	/api/viewcreator/views	データ参照定義を新規登録します。
PUT	/api/viewcreator/views	データ参照定義を更新します。
DELETE	/api/viewcreator/views/{dataViewCode}	データ参照を削除します。
GET	/api/viewcreator/views/{dataViewCode}	データ参照定義を取得します。
GET	/api/viewcreator/views_count	データ参照定義の検索を行い、件数を返却します。
GET	/api/viewcreator/views_pagination	データ参照定義の検索を行い、一覧用のデータ参照定義情報の配列を返却します。

「ExternalDataSource」をクリックするとViewCreatorで利用するIM-LogicDesignerのロジックフロー定義のメタ情報アクセスに関するAPIの一覧が表示されます。

**ExternalDataSource : CAP.Z.IWP.VIEWCREATOR.REST.DESCRPTION.TAG.EXTERNAL.DATA.SOURCE** Show/Hide | List Operations | Expand Operations

DELETE	/api/viewcreator/external_data_source/{dataType}
GET	/api/viewcreator/external_data_source/{dataType}
DELETE	/api/viewcreator/external_data_source/{dataType}/{dataId}
POST	/api/viewcreator/external_data_source/{dataType}/{dataId}

「Query」をクリックするとクエリに関するAPIの一覧が表示されます。

**Query** : クエリに関するAPIです。

Show/Hide | List Operations | Expand Operations

GET	/api/viewcreator/constants/query	クエリ関連の定数値配列を取得します。
POST	/api/viewcreator/db/execute	SQLの実行を行います。
GET	/api/viewcreator/db/fields	フィールド情報の一覧を取得します。
POST	/api/viewcreator/db/sql-fields	SELECT文からフィールド情報の一覧を取得します。
GET	/api/viewcreator/db/tables	テーブル情報の一覧を取得します。
GET	/api/viewcreator/db/views	ビュー情報の一覧を取得します。
GET	/api/viewcreator/ds/fields	拡張データソースのリソースで定義されているフィールド一覧を取得します。
GET	/api/viewcreator/ds/resources	拡張データソースのリソース一覧を検索します。
DELETE	/api/viewcreator/queries	クエリを一括削除します。
GET	/api/viewcreator/queries	クエリの定義情報を検索して配列で取得します。
POST	/api/viewcreator/queries	クエリを新規登録します。
PUT	/api/viewcreator/queries	クエリを更新します。
DELETE	/api/viewcreator/queries/{queryCode}	クエリ定義を削除します。
GET	/api/viewcreator/queries/{queryCode}	クエリ定義情報を取得します。
GET	/api/viewcreator/queries/{queryCode}/execute	指定されたクエリコードを持つクエリ定義から生成されるSQL文(SELECT文)を実行して結果を取得します。
GET	/api/viewcreator/queries/{queryCode}/execute/{startIndex}/{length}	指定されたクエリコードを持つクエリ定義から生成されるSQL文(SELECT文)を実行して結果を取得します。
GET	/api/viewcreator/queries/{queryCode}/sql	指定されたクエリコードを持つクエリ定義からSQL文(SELECT文)を生成・取得します。
GET	/api/viewcreator/queries_count	クエリの定義情報を検索してクエリ件数を取得します。
GET	/api/viewcreator/queries_pagination	クエリの定義情報を検索して一覧用クエリ定義情報の配列で取得します。
POST	/api/viewcreator/query_execute	指定されたクエリ定義から生成されるSQL文(SELECT文)を実行して結果を取得します。
POST	/api/viewcreator/query_execute/{startIndex}/{length}	指定されたクエリ定義から生成されるSQL文(SELECT文)を実行して結果を取得します。
POST	/api/viewcreator/query_sql	指定されたクエリ定義からSQL文(SELECT文)を生成・取得します。

「Route」をクリックするとクエリルーティングに関するAPIの一覧が表示されます。

**Route** : クエリルーティングに関するAPIです。

Show/Hide | List Operations | Expand Operations

GET	/api/viewcreator/query/query_route/routes	検索条件と一致するルーティング情報一覧を取得します。
POST	/api/viewcreator/query/query_route/routes	ルーティング情報を登録します。
DELETE	/api/viewcreator/query/query_route/routes/{routeId}	ルーティング情報を削除します。
GET	/api/viewcreator/query/query_route/routes/{routeId}	ルーティング情報を取得します。
PUT	/api/viewcreator/query/query_route/routes/{routeId}	ルーティング情報を更新します。
PUT	/api/viewcreator/query/query_route/routes_sort	ルーティング情報のソート番号を更新します。
GET	/api/viewcreator/query/query_router/initialize	ルーティング情報を初期化します。

例えば「GET /api/viewcreator/queries/{queryCode}」では、クエリコードからクエリ定義情報を取得できます。

GET /api/viewcreator/queries/{queryCode} クエリ定義情報を取得します。

**Implementation Notes**  
無効なクエリコードが指定された場合はエラーが返却されます。

**Response Class (Status default)**  
Model | Model Schema

```

{
  "error": true,
  "errorMessage": "string",
  "data": {
    "code": "string",
    "name": "string",
    "info": "string",
    "dataSource": "string",
    "tables": [
      {
        "code": "string"
      }
    ]
  }
}

```

Response Content Type

**Parameters**

Parameter	Value	Description	Parameter Type	Data Type
queryCode	5i7o4o3kli1pj	クエリコード	path	string

実行結果は以下の通りです。

**Response Body**

```

{
  "error": false,
  "data": {
    "code": "5i7o4o3kli1pj",
    "name": "日本のデータ",
    "info": "",
    "dataSource": "default",
    "tables": [
      {
        "code": "tcd00001",
        "name": "sample_population",
        "y": 204,
        "x": 243,
        "width": 191,
        "height": 143
      },
      {
        "code": "tcd00002",
        "name": "sample_prefecture",
        "y": 273,
      }
    ]
  }
}

```

**Response Code**  
200

**Response Headers**

```

{
  "cache-control": "no-store",
  "content-length": "1991",
  "content-type": "application/json; charset=UTF-8",
  "date": "Wed, 28 Oct 2020 05:40:40 GMT",
  "pragma": "no-cache",
  "server": "Resin/4.0.64"
}

```

## ルーティングの作成

クエリにアクセスするURLや認可など、ユーザがクエリを実行するための定義情報である「ルーティング定義」を作成します。作成したルーティングは、IM-BloomMakerやスクラッチ開発によるアプリケーションで利用できます。

## コラム

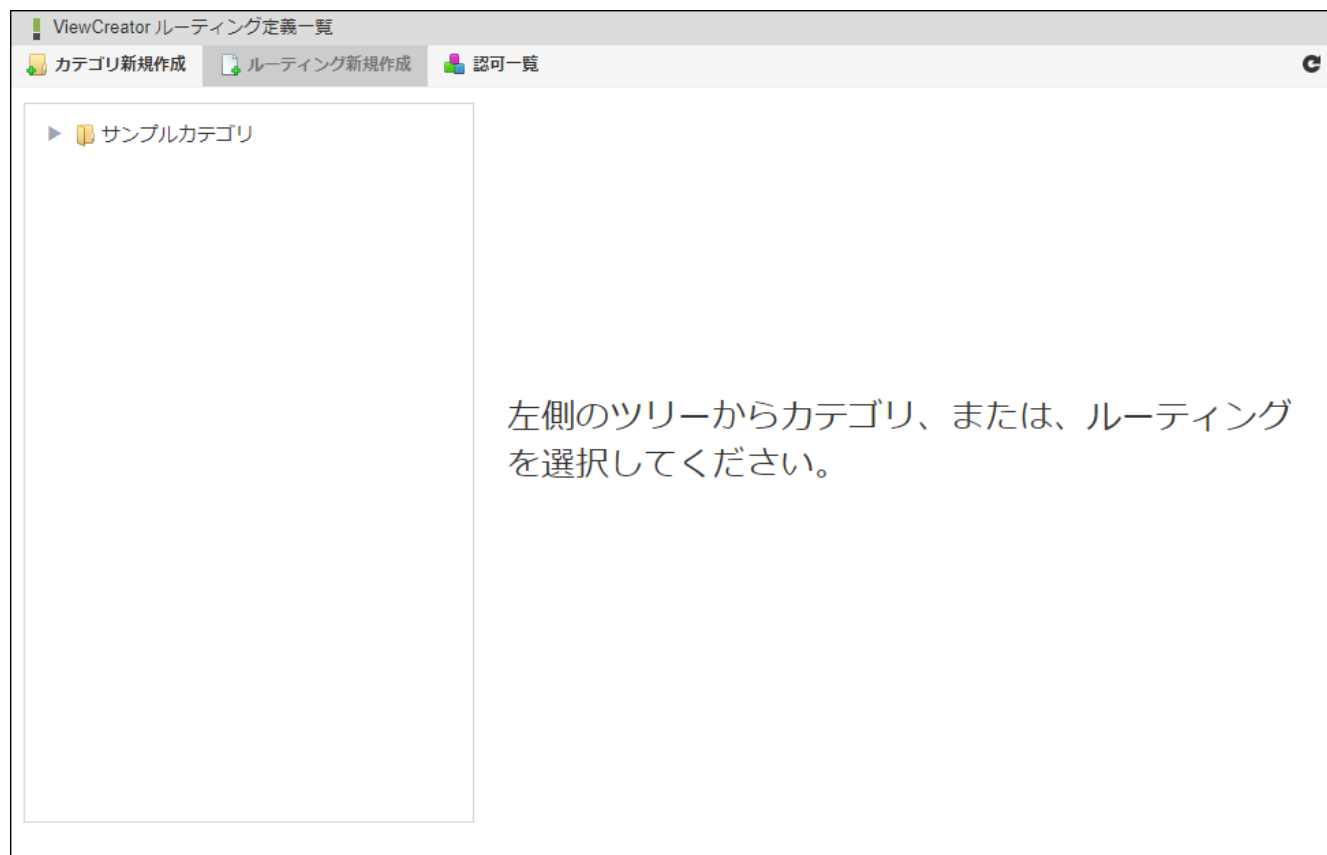
ルーティング定義作成機能は 2020 Winter(Azalea) から利用可能です。

### 目次

- ルーティング定義一覧画面
- ルーティングカテゴリの作成
- ルーティングの作成
  - ルーティングのクエリ設定
  - ルーティングの認証方式
  - ルーティングの認可設定
  - ルーティングの実行確認
- REST API 仕様
  - セッション管理
  - リクエストパラメータ
  - レスポンスヘッダ
  - レスポンスボディ
  - REST APIの仕様確認
- IM-BloomMakerを使用した画面作成例
  - 実行するクエリの作成
  - ルーティング定義の作成
  - 画面の作成

## ルーティング定義一覧画面

「サイトマップ」→「ViewCreator」→「ルーティング定義一覧」をクリックします。



図：「ルーティング定義一覧」画面

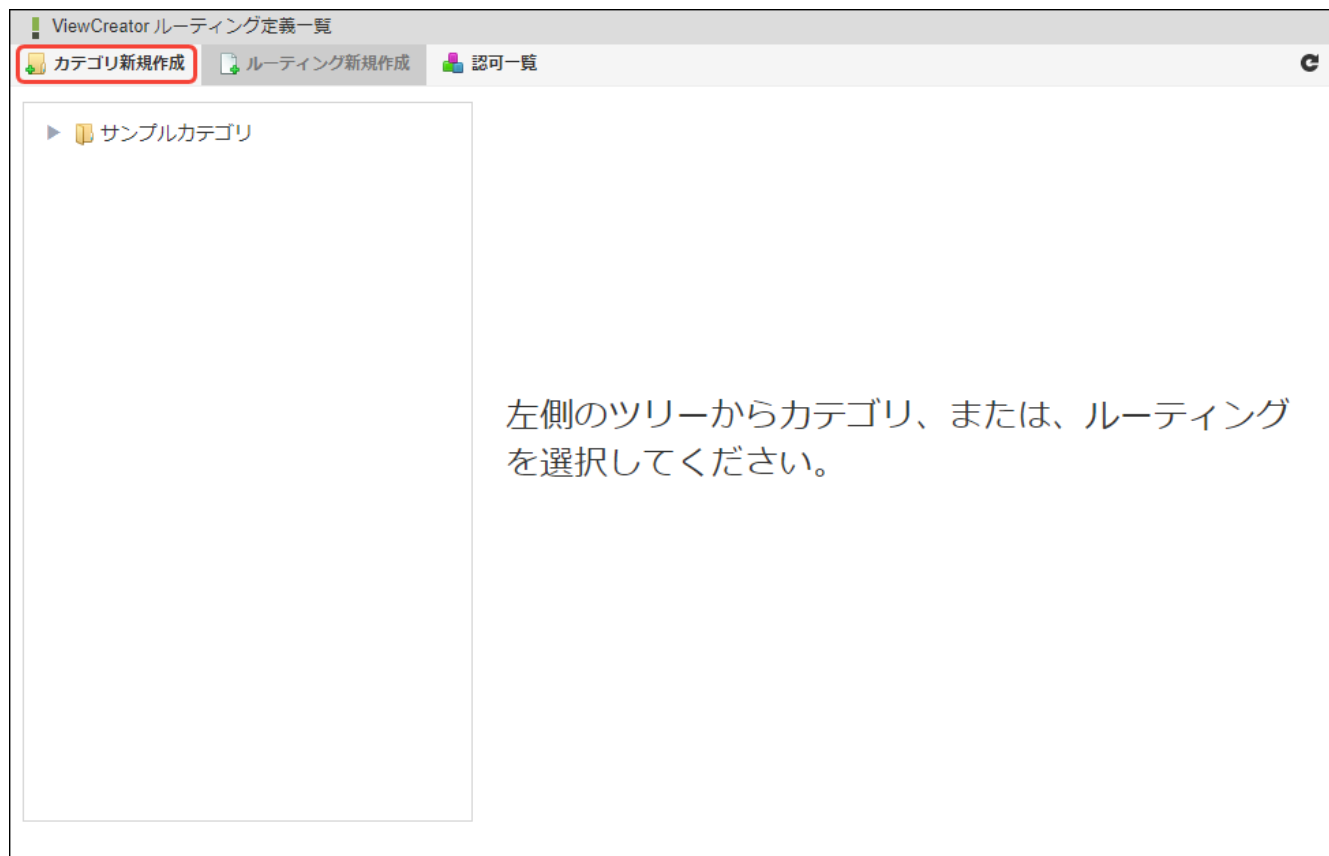
項目名	説明
カテゴリ新規作成	ルーティングカテゴリを新規に作成します。 詳細は「 <a href="#">ルーティングカテゴリの作成</a> 」を参照してください。

項目名	説明
ルーティング新規作成	ルーティングを新規に作成します。 詳細は「 <a href="#">ルーティングの作成</a> 」を参照してください。
認可一覧	「認可設定」画面を表示します。

## ルーティングカテゴリの作成

ルーティングはルーティングカテゴリに紐づけられる形で管理されているため、はじめにルーティングカテゴリを作成します。

1. 「ルーティング定義一覧」画面左上の「カテゴリ新規作成」をクリックします。



図：「ルーティング定義一覧」画面 - 「カテゴリ新規作成」

2. ルーティングツリーに新規のルーティングカテゴリが作成され、ルーティングカテゴリ情報が表示されます。



ViewCreator ルーティング定義一覧

カテゴリ新規作成 ルーティング新規作成 認可一覧

サンプルカテゴリ  
新しいカテゴリ

**基本**

カテゴリは第一階層にのみ作成できます。  
作成したカテゴリは、ツリーアイテムをドラッグアンドドロップすることで表示順序を変更できます。

カテゴリID \* 8fs2guqcoiv8o

カテゴリ名 標準 \* 新しいカテゴリ +

登録

図：ルーティングカテゴリ情報

項目名	必須	説明
カテゴリID	必須	カテゴリを一意に表す文字列を設定します。 新規作成時のみ設定可能です。登録後の変更はできません。
カテゴリ名	標準のみ必須	カテゴリを表す名称を設定します。 多言語に対応しています。

### **i** コラム

多言語可能な項目は、ロケールによって表示内容が変化します。  
標準・日本語が設定されている場合、以下の通り表示されます。

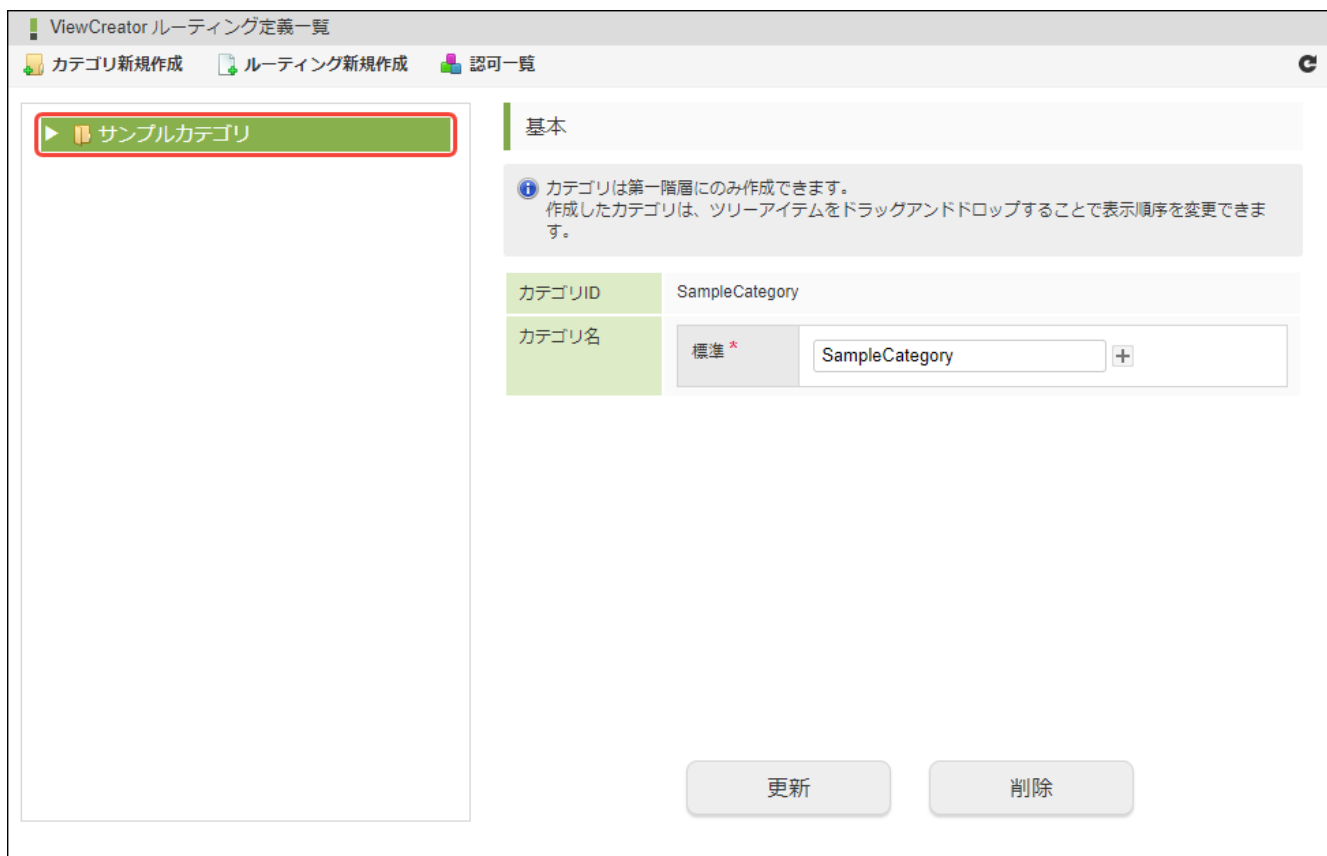
- 日本語 (ja) ロケールが設定されているユーザの場合、「日本語」に設定したものが表示されます。
- 日本語 (ja) ロケール以外が設定されているユーザの場合、「標準」に設定したものが表示されます。

- ルーティングカテゴリ作成に必要な情報を入力します。
- 「登録」をクリックします。

## ルーティングの作成

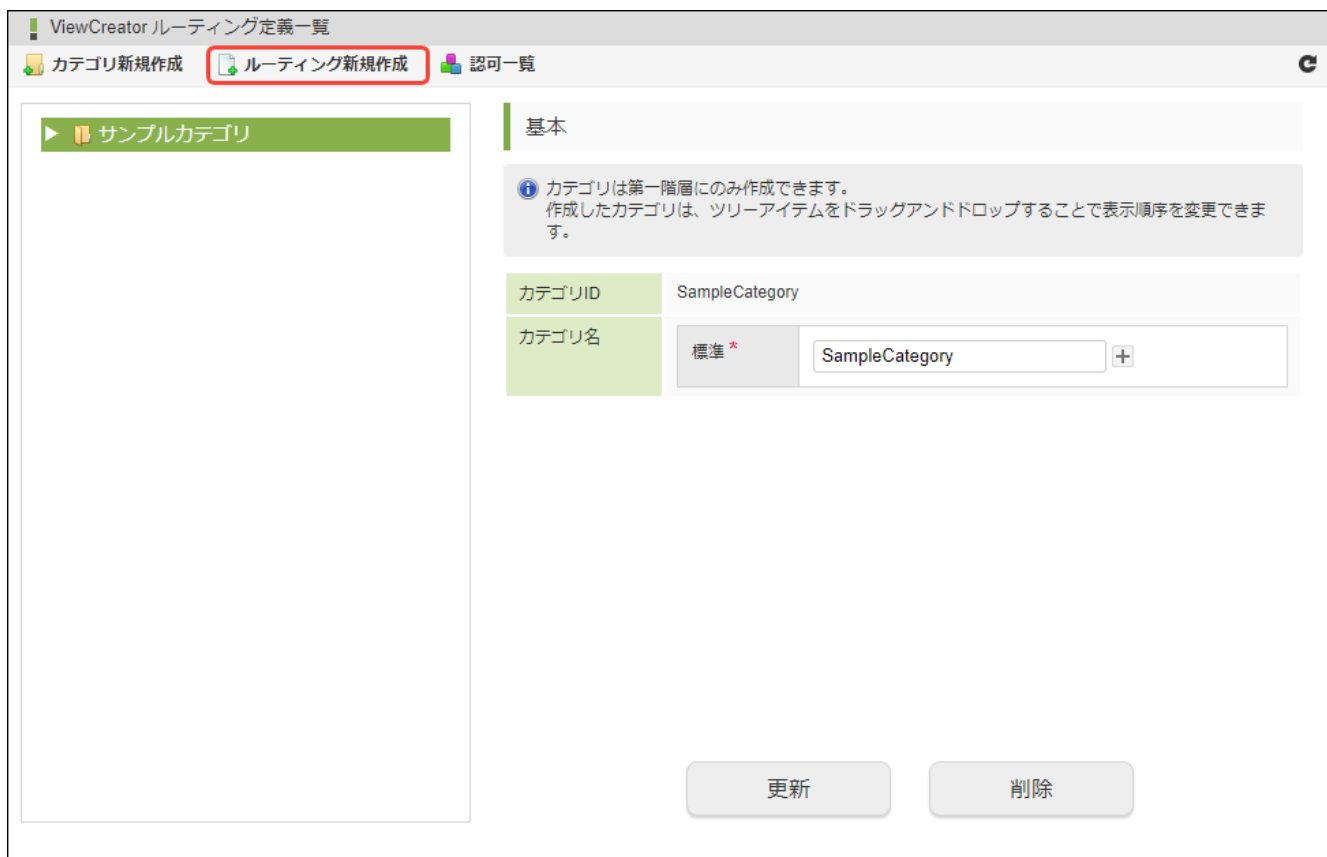
次に、ユーザがクエリを実行するための定義情報である「ルーティング」を作成します。

- 追加先のルーティングカテゴリをルーティングツリーから選択しクリックします。



図：「ルーティング定義一覧」画面 - ルーティングツリー

2. 「ルーティング定義一覧」画面左上の「ルーティング新規作成」をクリックします。



図：「ルーティング定義一覧」画面 - 「ルーティング新規作成」

3. ルーティングツリーに新規のルーティングが作成され、ルーティング情報が表示されます。

ViewCreator ルーティング定義一覧

カテゴリ新規作成 ルーティング新規作成 認可一覧

サンプルカテゴリ

サンプルルーティング

8 新しいルーティング

**基本**

① ルーティングはカテゴリの下にのみ作成できます。  
作成したルーティングは、ツリーアイテムをドラッグアンドドロップすることで表示順序や所属するカテゴリを変更できます。

カテゴリID SampleCategory

ルーティングID \* 8fscfpwo0uxv

ルーティングURL \* /imart/

対象クエリ \*

クエリコード

クエリ名

メソッド \*  GET  POST

認証方式 \*  IMAuthentication  OAuth  Basic

認可URI viewcreator-query-rest://8fscfpwo0uxv

レスポンス種別 \*  JSONに変換して返却  CSVとして返却

ルーティング名 標準 \* 新しいルーティング +

備考 標準  +

登録

図：ルーティング情報

項目名	必須	説明
カテゴリID	—	カテゴリIDを表示します。 登録内容の変更はできません。
ルーティングID	必須	ルーティングを一意に表す文字列を設定します。 新規作成時のみ設定可能です。登録後の変更はできません。
ルーティングURL	必須	ルーティングを呼び出す際のURLを設定します。 他のルーティングと重複するURLは設定できません。 新規作成時のみ設定可能です。登録後の変更はできません。
対象クエリ	必須	実行対象のクエリを設定します。 「検索」をクリックして実行するクエリを選択してください。 詳細は「 <a href="#">ルーティングのクエリ設定</a> 」を参照してください。
メソッド	必須	ルーティングを呼び出す際に指定するHTTPメソッドを設定します。 新規作成時のみ設定可能です。登録後の変更はできません。
認証方式	必須	ルーティングの認証方式を設定します。 各認証方式の詳細は「 <a href="#">ルーティングの認証方式</a> 」を参照してください。
OAuth スコープ	—	OAuth認証を使用する場合のスコープを設定します。 認証方式で OAuth を選択した場合のみ設定できます。

項目名	必須	説明
認可URI	—	認可URIを表示します。登録内容の変更はできません。 登録済みのルーティングの場合は、「認可設定」アイコンからルーティングの認可を設定できます。 詳細は「 <a href="#">ルーティングの認可設定</a> 」を参照してください。
レスポンス種別	—	ルーティングを実行した際に得られる出力データの形式を設定します。 各出力データの形式の詳細は「 <a href="#">レスポンスボディ</a> 」を参照してください。
ルーティング名	標準のみ必須	ルーティングを表す名称を設定します。 多言語に対応しています。
備考	—	ルーティングの補足情報を設定します。 多言語に対応しています。

### コラム

多言語可能な項目は、ロケールによって表示内容が変化します。  
標準・日本語が設定されている場合、以下の通り表示されます。

- 日本語 (ja) ロケールが設定されているユーザの場合、「日本語」に設定したものが表示されます。
- 日本語 (ja) ロケール以外が設定されているユーザの場合、「標準」に設定したものが表示されます。

### 注意

- ルーティングURLに指定できる文字列は「a-z」、「A-Z」、「0-9」、「- (ハイフン)」、「\_ (アンダースコア)」、「/ (スラッシュ)」、「. (ピリオド)」のいずれかです。
- ルーティングURLを「/ (スラッシュ)」から始めることはできません。

4. ルーティング作成に必要な情報を入力します。
5. 「登録」をクリックします。

## ルーティングのクエリ設定

実行するクエリは「クエリ検索」画面で一覧から選択できます。

1. ルーティング情報の対象クエリの「検索」アイコンをクリックすると、「クエリ検索」画面が表示されます。



図：「クエリ検索」画面

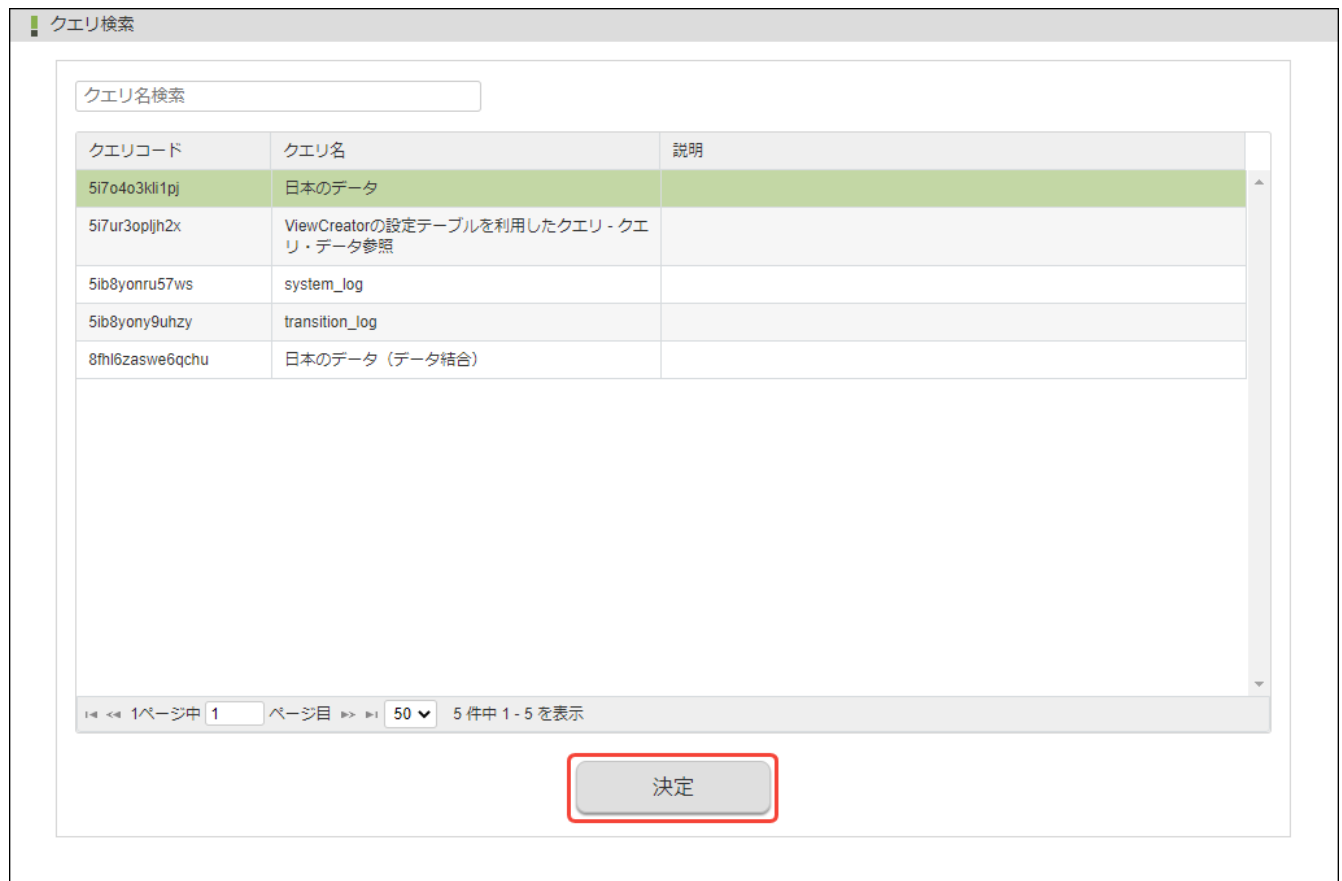
項目名	説明
クエリ名検索	一覧に表示するクエリを、クエリ名で絞り込めます。
決定	設定するクエリを決定します。 一覧でクエリが選択されていない場合は、クリックできません。



### コラム

クエリ一覧のページ毎の表示件数の選択候補は、設定ファイルの「query-list-settingsタグ」で設定できます。設定方法については「[intra-mart Accel Platform 設定ファイルリファレンス](#)」の「[クエリ一覧のリスト設定](#)」を参照してください。

- クエリを選択して「決定」をクリックすることで、ルーティング定義一覧画面のルーティング情報に反映されます。



図：「クエリ検索」画面 - クエリ選択

## ルーティングの認証方式

ルーティングの認証方式により、認証処理を行います。認証方式は以下が存在します。

- [IMAuthentication](#)
- [OAuth](#)
- [Basic](#)

### IMAuthentication

特別な認証処理を行わず、現在のCookieに紐づくセッションの認証状態のままクエリへのアクセスを実行します。

### OAuth

OAuth認証処理を行い、クエリへのアクセスを実行します。

この認証方式を利用したルーティングを実行する場合は、ヘッダにアクセストークンを付与してリクエストを送信してください。

```
GET /<resource_path> HTTP/1.1
Host: localhost
Authorization: Bearer <access_token>
```

アクセストークンの取得方法については以下のドキュメントを参照してください。

- [OAuth認証モジュール 仕様書](#)
- [OAuth 管理者操作ガイド](#)
- [OAuth プログラミングガイド](#)

### Basic

Basic認証処理を行い、クエリへのアクセスを実行します。

## i コラム

バーチャルテナントによる複数テナントを利用している環境において、リクエスト情報を利用したテナント自動解決機能を利用していない場合のみ、Basic認証のユーザ名にテナントを指定できます。

上記環境におけるBasic認証のユーザ名の指定方法による挙動の違いは以下の通りです。

- 「ユーザコード」の指定  
デフォルトテナントに対して認証します。
- 「テナントID\ユーザコード」の指定  
指定したテナントに対して認証します。

## ルーティングの認可設定

作成したルーティングの認可設定は、初期状態では全ての対象者条件に対して「禁止」が設定されます。

そのため、作成したルーティングに対して認可設定をする必要があります。

1. 認可設定をするルーティングをルーティングツリーから選択します。
2. 「認可URI」の「認可設定」アイコンをクリックします。

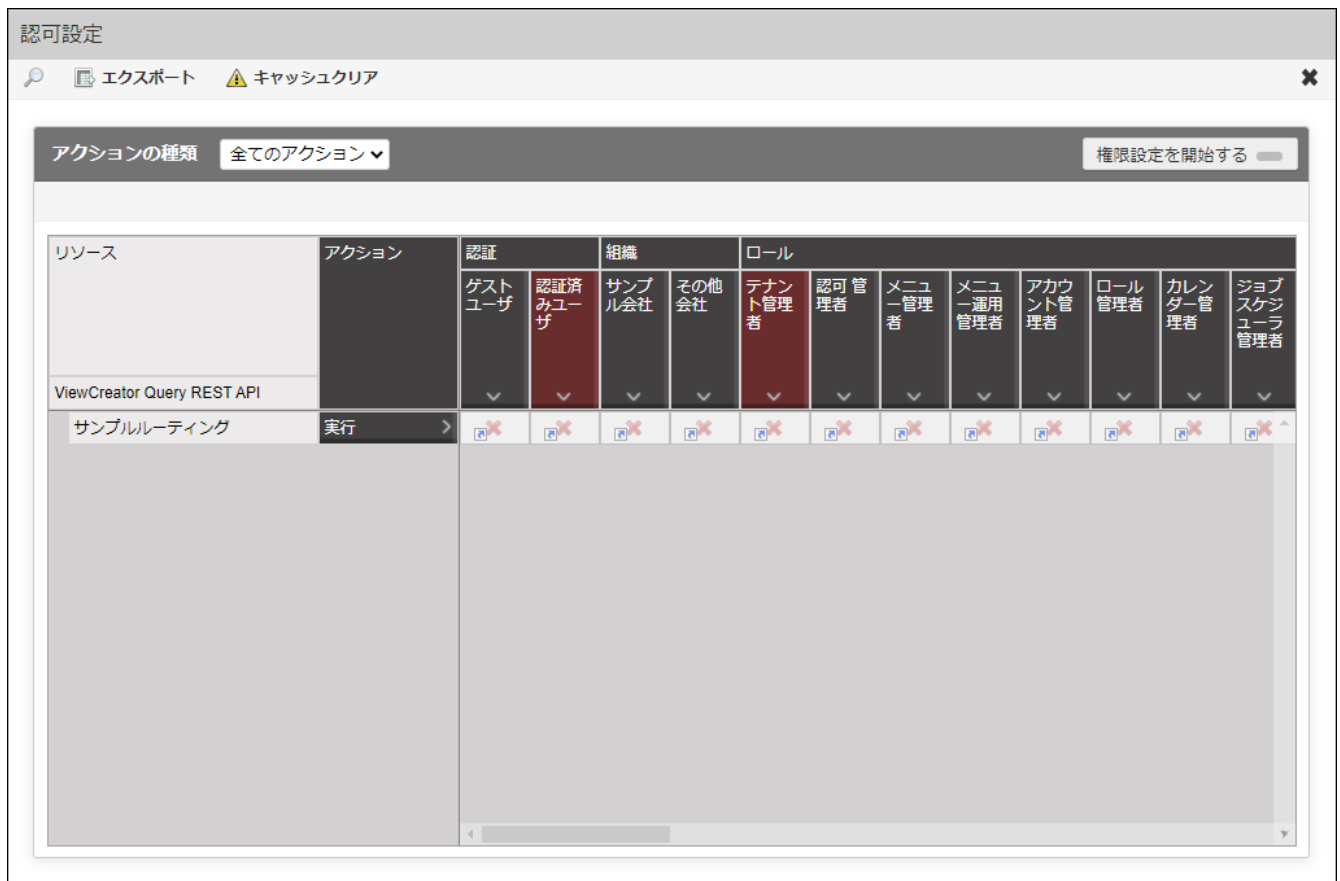
The screenshot shows the 'ViewCreator ルーティング定義一覧' (ViewCreator Routing Definition List) interface. On the left, a tree view shows 'サンプルカテゴリ' (Sample Category) expanded to 'サンプルルーティング' (Sample Routing). The main area displays the configuration for this routing. The '認可URI' (Authorization URI) field is highlighted with a red box, and a small gear icon (the '認可設定' icon) is visible next to it. The configuration includes:

- カテゴリID: SampleCategory
- ルーティングID: SampleRoute
- ルーティングURL: /imart/viewcreator/sample/routes/api/get
- 対象クエリ (Target Query):
  - クエリコード (Query Code): 5i7o4o3kii1pj
  - クエリ名 (Query Name): 日本のデータ
- メソッド (Method): GET (selected), POST
- 認証方式 (Authentication Method): IMAuthentication (selected), OAuth, Basic
- 認可URI (Authorization URI): viewcreator-query-rest://SampleRoute
- レスポンス種別 (Response Type): JSONに変換して返却 (selected), CSVとして返却

Buttons at the bottom include '更新' (Update), '削除' (Delete), and 'OpenAPI Specification'.

図：「認可URI」 - 「認可設定」アイコン

3. 「認可設定」画面が表示されます。



図：「認可設定」画面

4. 認可設定を行います。認可の設定方法は「[Intra-mart Accel Platform 認可仕様書](#)」の「[認可設定画面](#)」を参照してください。

#### 注意

認可処理を行う権限を持たないユーザが「認可」アイコンをクリックした場合、警告メッセージが表示されます。メッセージが表示された場合、操作したユーザに対して適切な権限が付与されているか確認してください。

#### ルーティングの実行確認

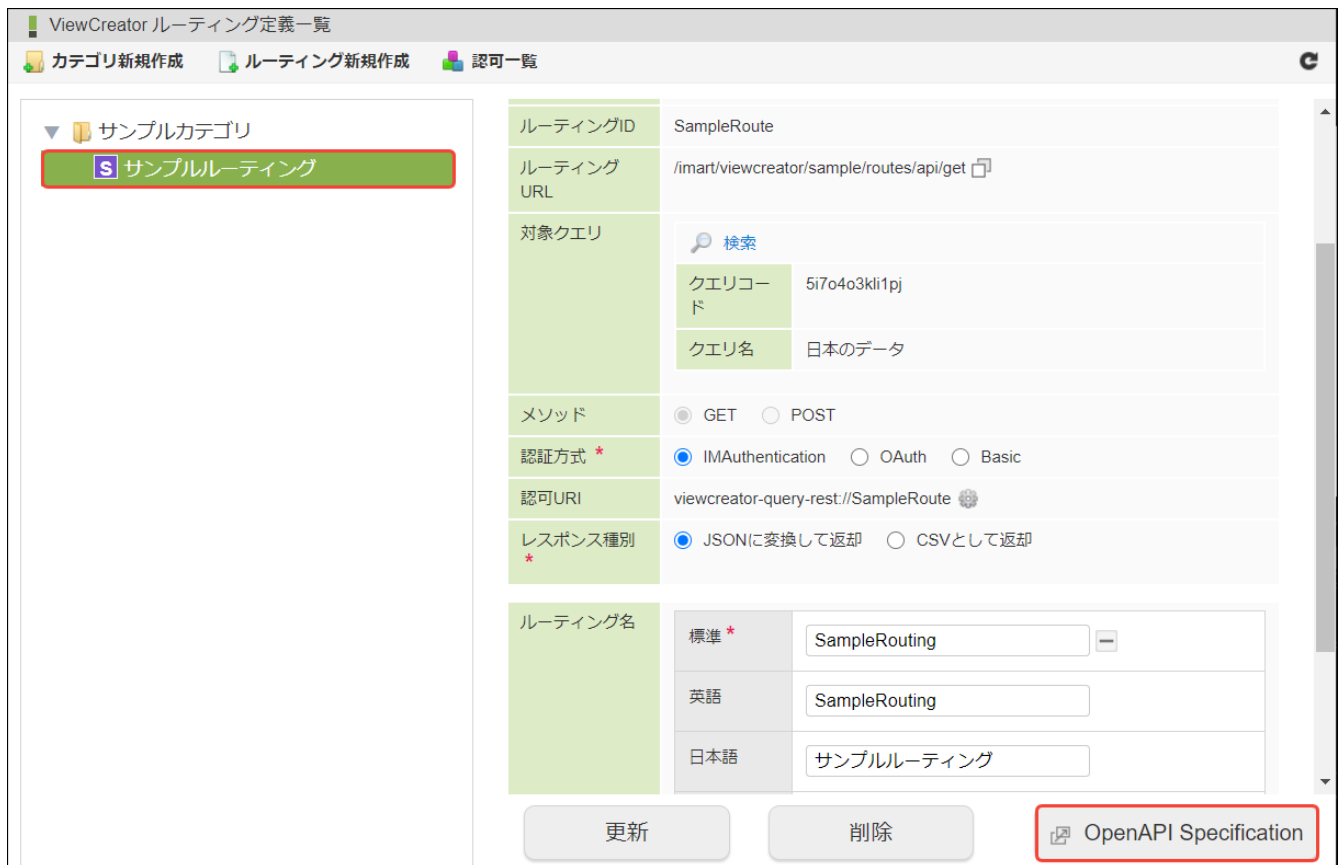
作成したルーティングは、OpenAPIを利用することで実行できます。

クエリの実行はREST APIを介して実行されます。

REST APIの詳細は「[REST API 仕様](#)」を参照してください。

- ルーティング定義一覧画面では、ルーティング定義をもとに、OpenAPI上で対象のルーティングを実行するためのページへのリンクを提供します。





図：「ルーティング定義一覧」画面 - OpenAPIを開く



図：OpenAPI REST APIビューワー

GET /viewcreator/sample/routes/api/get?metadata メタデータ取得API

実行するクエリのカラム定義や抽出条件値定義を取得します。

**Parameters** Cancel

No parameters

**Execute** **Clear**

**Responses**

**Curl**

```
curl -X GET "http://localhost:8080/imart/viewcreator/sample/routes/api/get?metadata" -H "accept: */*"
```

**Request URL**

```
http://localhost:8080/imart/viewcreator/sample/routes/api/get?metadata
```

**Server response**

Code	Details
200	<p><b>Response body</b></p> <pre>{   "columns": [     {       "code": "ccd00001",       "name": "year",       "caption": "年度",       "type": "NUMBER"     }   ] }</pre>

図 : REST API実行 - クエリのメタ情報取得

GET /viewcreator/sample/routes/api/get クエリ実行API

クエリを実行して結果を取得します。

**Parameters** Cancel

No parameters

Execute Clear

**Responses**

Curl

```
curl -X GET "http://localhost:8080/imart/viewcreator/sample/routes/api/get" -H "accept: */*"
```

Request URL

```
http://localhost:8080/imart/viewcreator/sample/routes/api/get
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "data": [     {       "year": "1970",       "name": "北海道",       "ccd00003": "北海道",       "age": "0 - 14歳",       "area": "83455.33",       "population": "1309"     }   ], }</pre>

図：REST API実行 - クエリの実行結果取得

## REST API 仕様

クエリを実行するREST APIの動作仕様について説明します。

### セッション管理

REST APIではリクエストヘッダ *X-Intramart-Session* を指定することによりセッション管理を行えます。

*X-Intramart-Session* には、*keep*、*once*、*never* の3つの値が利用できます。

ヘッダを省略した場合は *once* を指定した場合と同じです。

指定した値による挙動は Web API Maker と同様です。詳しい挙動については以下を参照してください。

- セッション管理

### リクエストパラメータ

リクエストパラメータを設定することで、クエリの実行条件をカスタマイズできます。

パラメータ	データ型	説明
<i>limit</i>	数値型	<p>最大取得レコード件数を設定します。</p> <p>省略した場合は、デフォルト設定（100件）が適用されます。</p> <p>0を設定した場合は全件取得します。</p> <p>設定例: <i>limit=10</i></p>

パラメータ	データ型	説明
<i>offset</i>	数値型	レコードの取得開始位置を設定します。 省略した場合は、デフォルト設定 (0) が適用されます。 設定例: <i>offset=10</i>
<i>columns</i>	文字列型	取得対象とするカラムコードまたはカラム名をカンマ区切りで設定します。 省略した場合は、クエリで定義されているすべてのカラムが取得対象です。 カラムコードおよびカラム名は大文字・小文字を区別します。 設定例: <i>columns=user_cd,ccd00003,user_name</i>
<i>sort</i>	文字列型	第1ソートキーとするカラムコードまたはカラム名を設定します。 省略可能です。クエリで定義されているソート順に対して、条件を追加または差し替える場合に設定します。 <i>order</i> パラメータと組み合わせて使用します。 カラムコードおよびカラム名は大文字・小文字を区別します。 設定例: <i>sort=user_cd&amp;order=asc</i>
<div style="background-color: #e1f5fe; padding: 10px; border: 1px solid #cfcfcf;"> <p> <b>コラム</b></p> <p><i>order</i> パラメータを設定しないことで、クエリで定義されているソート条件を除去できます。</p> </div>		
<i>order</i>	文字列型	第1ソートキー ( <i>sort</i> ) に対するソート方向を設定します。 <i>asc</i> または <i>desc</i> を設定します。 設定例: <i>sort=user_cd&amp;order=desc</i>
<i>secondSort</i>	文字列型	第2ソートキーとするカラムコードかカラム名を設定します。 詳細は <i>sort</i> パラメータの説明を参照してください。
<i>secondOrder</i>	文字列型	第2ソートキー ( <i>secondSort</i> ) に対するソート方向を設定します。 詳細は <i>order</i> パラメータの説明を参照してください。
<i>thirdSort</i>	文字列型	第3ソートキーとするカラムコードかカラム名を設定します。 詳細は <i>sort</i> パラメータの説明を参照してください。
<i>thirdOrder</i>	文字列型	第3ソートキー ( <i>thirdSort</i> ) に対するソート方向を設定します。 詳細は <i>order</i> パラメータの説明を参照してください。
<i>responseKeyType</i>	文字列型	レスポンスデータのキー (JSON形式の場合キー、CSV形式の場合ヘッダ) になる項目を設定します。 以下のいずれかを設定できます。 <ul style="list-style-type: none"> <li>▪ <b>NAME</b> カラム名 (物理名) をレスポンスデータのキーに設定します。</li> <li>▪ <b>CODE</b> カラムコードをレスポンスデータのキーに設定します。</li> <li>▪ <b>CAPTION</b> カラムの論理名をレスポンスデータのキーに設定します。</li> </ul> <p>省略した場合は、デフォルト設定 (<b>NAME</b>) が適用されます。 設定例: <i>responseKeyType=NAME</i></p>
<div style="background-color: #fff9c4; padding: 10px; border: 1px solid #ccc;"> <p> <b>注意</b></p> <p><b>CODE</b> 以外はキーが重複する可能性があり、重複した場合、後のキーにはカラムコードが使用されます。</p> </div>		
<i>metadata</i>	—	クエリ定義のメタ情報を取得する場合に設定します。 クエリの実行結果ではなく、メタ情報がレスポンスデータに設定されます。 詳細は「 <a href="#">レスポンスボディ</a> 」を参照してください。

### コラム

動的パラメータを使用することで、リクエストパラメータに抽出条件を設定できます。  
詳細は「[動的パラメータ](#)」の「`<%REQUEST_PARAMETER%>` タグ」を参照してください。

## レスポンスヘッダ

クエリの実行が正常に完了した場合に返却されるレスポンスヘッダは以下の通りです。

パラメータ	データ型	説明
<code>x-jp-co-intra-mart-viewcreator-total-count</code>	数値型	クエリの実行結果の総レコード件数が格納されます。 <code>limit</code> パラメータや <code>offset</code> パラメータにより、返却されるデータ件数と異なる場合があります。

## レスポンスボディ

クエリ実行によりレスポンスボディに格納されるデータは、ルーティング定義に設定されているレスポンス種別により異なります。

- JSONに変換して返却

レスポンスボディにはJSON形式のデータが格納されます。

```
{
  "data": [
    {
      "fieldName1": "field1Value1",
      "fieldName2": "field2Value1",
      "fieldName3": "field3Value1"
    },
    {
      "fieldName1": "field1Value2",
      "fieldName2": "field2Value2",
      "fieldName3": "field3Value2"
    }
  ],
  "totalCount": 2
}
```

<code>data</code>	実行結果の一覧です。 レコードを表すオブジェクトの配列で構成されます。
<code>totalCount</code>	実行結果の総件数です。 レスポンスヘッダに格納される <code>x-jp-co-intra-mart-viewcreator-total-count</code> パラメータと同じ値が格納されます。

- CSVとして返却

レスポンスボディにはCSV形式のデータが格納されます。  
1行目にはCSVヘッダ、2行目以降はレコードデータが格納されます。

```
fieldName1,fieldName2,fieldName3
field1Value1,field2Value1,field3Value1
field1Value2,field2Value2,field3Value2
```



#### コラム

CSV形式では、データがnull値の場合は空文字を出力します。

- メタ情報の返却

クエリ定義のメタ情報を取得した場合は、レスポンスボディにJSON形式のメタ情報が格納されます。

```
{
  "columns": [
    {
      "code": "string",
      "name": "string",
      "type": "STRING",
      "caption": "string"
    }
  ],
  "conditions": [
    {
      "key": "string",
      "type": "STRING"
    }
  ]
}
```

<i>columns</i>	クエリ定義に含まれるカラム情報の一覧です。 <ul style="list-style-type: none"> <li>▪ <i>code</i> ccd + 5桁の連番で採番された、クエリ定義内においてユニークなコードです。</li> <li>▪ <i>name</i> DDL上で定義されたフィールドの物理名です。</li> <li>▪ <i>type</i> カラムのデータ型です。 "STRING", "NUMBER", "DATE", "TIMESTAMP", "BOOLEAN"のいずれかが格納されます。</li> <li>▪ <i>caption</i> 「クエリ編集」画面で設定したキャプションです。</li> </ul>
<i>conditions</i>	クエリ定義に含まれるリクエストパラメータの動的抽出条件 (<%REQUEST_PARAMETER(KEY_NAME)%>) の一覧です。 <ul style="list-style-type: none"> <li>▪ <i>key</i> パラメータ名 (KEY_NAME) が格納されます。</li> <li>▪ <i>type</i> 抽出条件が設定されているカラムのデータ型です。 "STRING", "NUMBER", "DATE", "TIMESTAMP", "BOOLEAN"のいずれかが格納されます。</li> </ul>

### コラム

クエリ実行により返却される日付型およびタイムスタンプ型のデータは、「JSONに変換して返却」、「CSVとして返却」いずれの場合も ISO 8601 形式の日時フォーマット文字列です（タイムゾーンには、システム・デフォルトのタイムゾーン（Java VM のタイムゾーン）が適用されます）。

例：2017-08-01T12:34:56.789Z

### 注意

バイナリ型のカラムはREST APIではサポートしません。

対象のクエリにバイナリ型のカラムが含まれている場合は、クエリの実行結果、メタ情報いずれもカラムを除外した結果を返却します。

## REST APIの仕様確認

次に、OpenAPIを利用してクエリ実行結果を取得するREST APIの仕様を確認します。

- [OpenAPIとは](#)
- [REST APIを実行する](#)

### OpenAPIとは

OpenAPIはソースコードやドキュメントなどにアクセスせずに、APIの機能を理解できるREST APIの標準インタフェースを定義しています。OpenAPI仕様はREST APIを記述するためのフォーマットです。

Swagger UIは、OpenAPI仕様で定義されたAPIのドキュメントを視覚的にレンダリングするツールです。

## コラム

OpenAPIについての詳細は、以下のリンクを参照してください。

- [OpenAPI Specification](#)
- [OpenAPI Specification\(Github\)](#)
- [Swagger UI\(Github\)](#)

## REST APIを実行する

- 仕様確認をするルーティングをツリーから選択し、「OpenAPI Specification」ボタンをクリックします。

The screenshot shows the 'ViewCreator ルーティング定義一覧' (ViewCreator Routing Definition List) interface. On the left, a tree view shows 'サンプルカテゴリ' (Sample Category) expanded to 'サンプルルーティング' (Sample Routing), which is highlighted with a red box. The main area displays the configuration for the selected routing:

- ルーティングID: SampleRoute
- ルーティングURL: /imart/viewcreator/sample/routes/api/get
- 対象クエリ: Search box with '検索' (Search) button. Below it, 'クエリコード' (Query Code) is 5i7o4o3kli1pj and 'クエリ名' (Query Name) is 日本 of データ (Data of Japan).
- メソッド: GET (selected) or POST
- 認証方式\*: IMAuthentication (selected) or OAuth or Basic
- 認可URI: viewcreator-query-rest://SampleRoute
- レスポンス種別\*: JSONに変換して返却 (selected) or CSVとして返却
- ルーティング名:
  - 標準\*: SampleRouting
  - 英語: SampleRouting
  - 日本語: サンプルルーティング

At the bottom, there are buttons for '更新' (Update), '削除' (Delete), and 'OpenAPI Specification' (highlighted with a red box).

図：REST API実行 - 「OpenAPI Specification」のクリック

- OpenAPIの提供するREST APIビューワーの画面が新しいウィンドウで表示されます。

図 : REST API実行 - OpenAPI REST APIビューワー

画面項目	説明
Info	APIに関するタイトルや説明、バージョンなどです。
Server	ターゲットサーバの接続情報です。
Tag	オペレーションに関連するタグ情報です。「クエリルーティング名」
Paths	利用可能なパスとオペレーションの情報です。

- クエリ実行APIの「GET」ボタンをクリックします。

図 : REST API実行 - 「GET」のクリック

- 「Try it out」ボタンをクリックし、実行準備段階に遷移します。デフォルト値が表示され、リクエストパラメータを入力できます。



**GET**
/viewcreator/sample/routes/api/get クエリ実行API

クエリを実行して結果を取得します。

Parameters
Try it out

Name	Description
limit <i>integer</i> <i>(query)</i>	最大取得レコード件数。 0 が指定された場合は全件取得します。 <i>Default value : 100</i>
offset <i>integer</i> <i>(query)</i>	レコード取得開始位置。 <i>Default value : 0</i>
columns <i>string</i> <i>(query)</i>	取得対象とするカラム名かカラムコードをカンマ区切りで指定します。 省略された場合は、クエリで定義されているすべてのカラムが取得対象です。 例) "year,name,ccd00003"
sort <i>string</i> <i>(query)</i>	第1ソートキーとするカラムコードかカラム名を指定します。
order <i>string</i> <i>(query)</i>	第1ソートキーで指定されたカラムに対するソート方向。 <i>Available values : asc, desc</i> <i>Default value : asc</i>

図：REST API実行 - 「Try it out」のクリック

画面項目	説明
Parameters	REST APIの要求するパラメータの詳細です。 ここでは以下が確認できます。詳細は「 <a href="#">リクエストパラメータ</a> 」を参照してください。 <ul style="list-style-type: none"> <li>■ パラメータ名、説明、タイプ</li> <li>■ データ形式、デフォルト値</li> </ul>
「Try it out」 ボタン	OpenAPI上から対象のREST APIの実行準備段階に遷移します。

5. 必要項目を入力し、「Execute」ボタンをクリックします。

入力例は以下の通りです。

- limit : 2
- offset : 0
- columns : age,year,area
- sort : age
- order : asc
- secondSort : year
- secondOrder : asc
- thirdSort : area
- thirdOrder : asc
- responseKeyType : NAME

order	第1ソートキーで指定されたカラムに対するソート方向。
string (query)	<input type="text" value="asc"/>
secondSort	第2ソートキーとするカラムコードかカラム名を指定します。
string (query)	<input type="text" value="year"/>
secondOrder	第2ソートキーで指定されたカラムに対するソート方向。
string (query)	<input type="text" value="asc"/>
thirdSort	第3ソートキーとするカラムコードかカラム名を指定します。
string (query)	<input type="text" value="area"/>
thirdOrder	第3ソートキーで指定されたカラムに対するソート方向。
string (query)	<input type="text" value="asc"/>
responseKeyType	レスポンスデータのキー（JSON形式の場合キー、CSV形式の場合ヘッダ）になる項目を指定します。 NAME カラム名（物理名） CODE カラムコード CAPTION カラムの論理名、説明 CODE以外はキーが重複する可能性があり、重複した場合、後のキーにはカラムコードが使用されます。
string (query)	<input type="text" value="NAME"/>
<input type="button" value="Execute"/>	

図：REST API実行 - 「Execute」のクリック

6. 実行結果が表示されます。

Curl

```
curl -X GET "http://localhost:8080/imart/viewcreator/sample/routes/api/get?limit=2&offset=0&columns=age%2Cyear%2Carea&sort=age&order=asc&secondSort=year&secondOrder=asc&thirdSort=area&thirdOrder=asc&responseKeyType=NAME" -H "accept: application/json"
```

Request URL

```
http://localhost:8080/imart/viewcreator/sample/routes/api/get?limit=2&offset=0&columns=age%2Cyear%2Carea&sort=age&order=asc&secondSort=year&secondOrder=asc&thirdSort=area&thirdOrder=asc&responseKeyType=NAME
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "data": [     {       "age": "0 - 14歳",       "year": "1970",       "area": "1862.01"     },     {       "age": "0 - 14歳",       "year": "1970",       "area": "1893.76"     }   ],   "totalCount": 987 }</pre> <p>Download</p> <p>Response headers</p> <pre>content-length: 130 content-type: application/json date: Tue 02 Mar 2021 01:00:44 GMT server: Resin/4.0.64 x-jp-co-intra-mart-viewcreator-total-count: 987</pre>

図：REST API実行 - 実行結果

画面項目	説明
Curl	Curlを用いて対象のREST APIを実行する場合のコマンドです。
Request URL	実行時にリクエストを送信したURLです。
Code	REST APIを実行した結果として返却されたHTTPステータスコードです。
Response body	レスポンスの結果（Body情報）です。各出力データの形式の詳細は「 <a href="#">レスポンスボディ</a> 」を参照してください。
Response headers	REST APIを実行した結果として返却されたレスポンスヘッダです。詳細は「 <a href="#">レスポンスヘッダ</a> 」を参照してください。

## IM-BloomMakerを使用した画面作成例

IM-BloomMakerを利用してクエリの実行結果一覧を表示する画面を作成する例を紹介します。  
検索条件を入力するエリアとクエリの実行結果を一覧で表示するエリアを配置した画面を作成します。

## ViewCreator ルーティングサンプル

年度

地域名

年度	地域名	都道府県名	年齢	面積	人口
1970	北海道	北海道	0 - 14歳		1309
1970	北海道	北海道	15 - 64歳		3576
1970	北海道	北海道	65歳以上		299
1970	東北	青森	0 - 14歳		397
1970	東北	福島	0 - 14歳		508
1970	東北	山形	0 - 14歳		288
1970	東北	秋田	0 - 14歳		303
1970	東北	宮城	0 - 14歳		449
1970	東北	岩手	0 - 14歳		362
1970	東北	山形	15 - 64歳		833

1 2 3 4 5 ... 99

図：完成イメージ

 注意

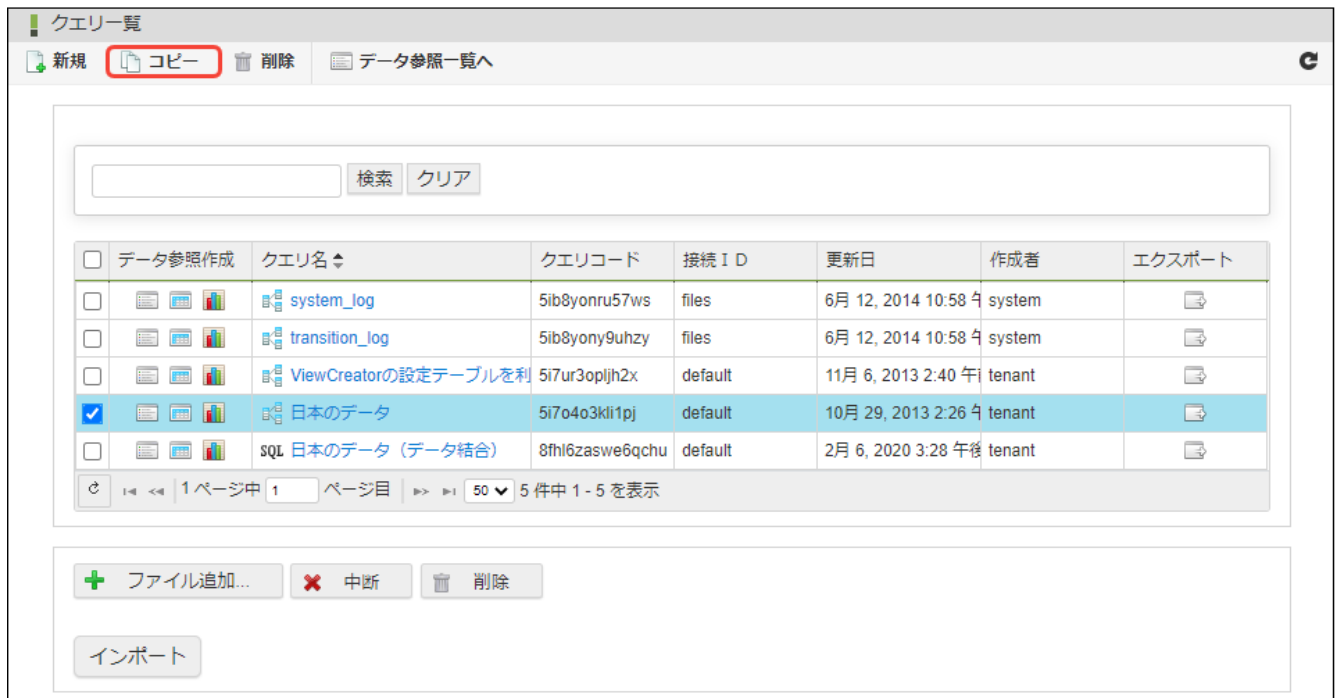
本作成例ではサンプルデータを使用するため、事前にテナント環境・サンプルデータセットアップが必要です。  
 サンプルデータの投入については「[intra-mart Accel Platform セットアップガイド](#)」の「[サンプルデータの投入](#)」を参照してください。

## 実行するクエリの作成

はじめに、実行するクエリを作成します。

本作成例では、サンプルデータのクエリを利用します。

1. 「サイトマップ」→「ViewCreator」→「クエリ一覧」から、「クエリ一覧」画面を表示します。
2. 一覧の「日本のデータ」にチェックを付け、「クエリ一覧」画面上部の「コピー」をクリックします。



図：「クエリー一覧」画面 - クエリのコピー

- 一覧からコピーしたクエリ「コピー～日本のデータ」をクリックし、「クエリー編集」画面を表示します。



図：「クエリー一覧」画面 - 「コピー～日本のデータ」

- 「クエリー編集」画面下部の「カラム一覧」タブの「キャプション」を以下のように編集します。

テーブル/ビュー/リソース	カラム/フィールド	キャプション
sample_population	year	year
sample_region	name	regionName
sample_prefecture	name	prefectureName
sample_age	age	age
sample_prefecture_area	area	area
sample_population	population	population

上 下	上 下	テーブルビュー/リソース	カラム/フィールド	型	ソート	キャプション	削除
↑	↓	sample_populati	year	数値	指定なし	year	✖
↑	↓	sample_region	name	文字列	指定なし	regionName	✖
↑	↓	sample_prefectu	name	文字列	指定なし	prefectureName	✖
↑	↓	sample_age	age	文字列	指定なし	age	✖
↑	↓	sample_prefectu	area	数値	指定なし	area	✖
↑	↓	sample_populati	population	数値	指定なし	population	✖

図：「クエリー一覧」画面 - キャプション設定

- 「クエリー編集」画面下部の「抽出条件一覧」タブをクリックします。

上 下	上 下	テーブルビュー/リソース	カラム/フィールド	型	ソート	キャプション	削除
↑	↓	sample_populati	year	数値	指定なし	year	✖
↑	↓	sample_region	name	文字列	指定なし	regionName	✖
↑	↓	sample_prefectu	name	文字列	指定なし	prefectureName	✖
↑	↓	sample_age	age	文字列	指定なし	age	✖
↑	↓	sample_prefectu	area	数値	指定なし	area	✖
↑	↓	sample_populati	population	数値	指定なし	population	✖

図：「クエリー編集」画面 - 「抽出条件一覧」タブ

- 「クエリー編集」画面のデザイナーで「sample\_population」テーブルの「year」フィールドをダブルクリックします（「抽出条件一覧」タブにカラムが追加されます）。

上 下	上 下	テーブルビュー/リソース	カラム/フィールド	抽出方法	条件値	削除
↑	↓	sample_populati	year	完全一致		✖

図：「クエリー編集」画面 - 抽出条件追加

7. 追加した抽出条件を以下のとおりに設定します。

- 抽出方法：「以上」
- 条件値：<%REQUEST\_PARAMETER(minimumYearParameter)%>

上 へ	下 へ	テーブル/ビュー/リソース	カラム/フィールド	抽出方法	条件値	削除
↑	↓	sample_population	year	以上	<%REQUEST_PARAMETER(minimumYearParameter)%>	✖

図：「クエリ編集」画面 - 抽出条件設定

8. 同様に、以下のとおりに抽出条件を追加します。

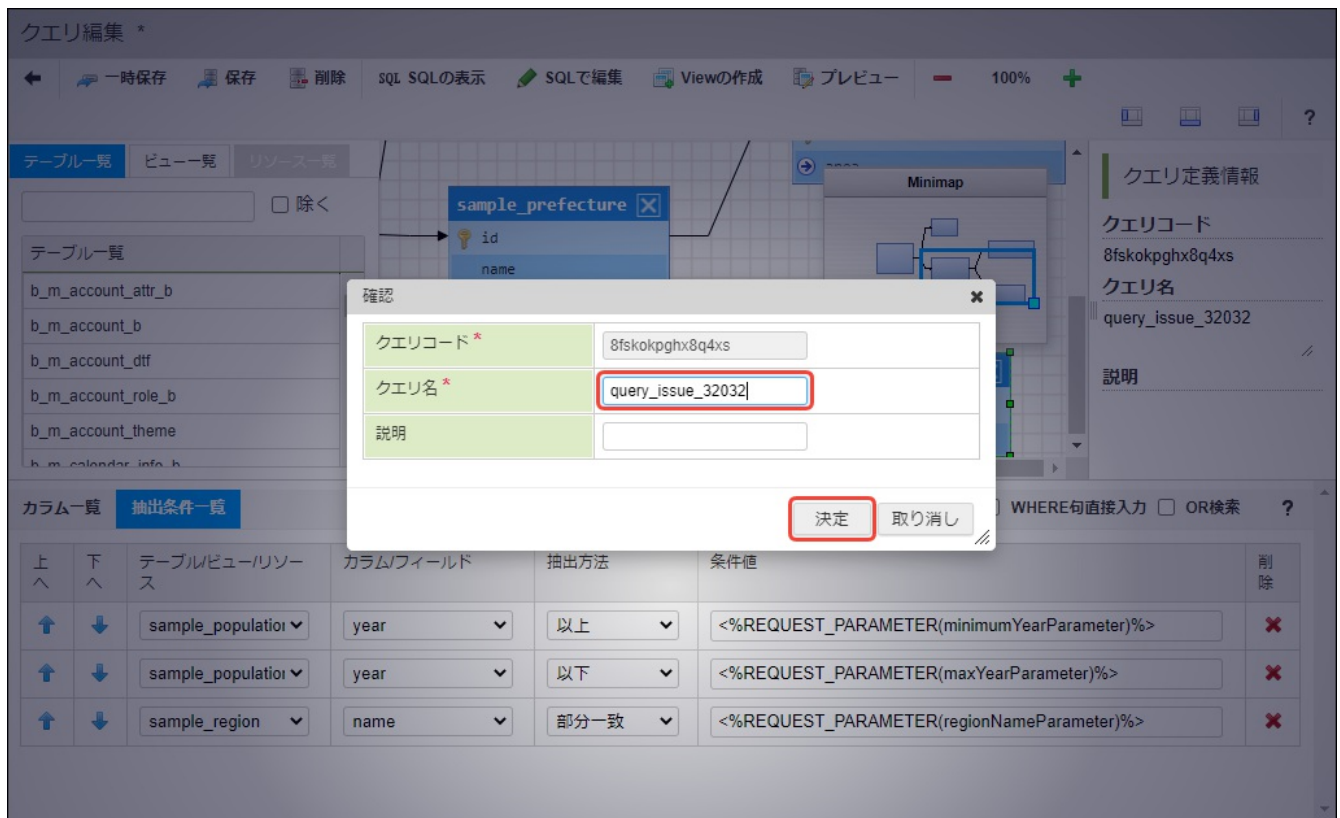
- テーブル/ビュー/リソース：sample\_population
- カラム/フィールド：year
- 抽出方法：「以下」
- 条件値：<%REQUEST\_PARAMETER(maxYearParameter)%>
- テーブル/ビュー/リソース：sample\_region
- カラム/フィールド：name
- 抽出方法：「部分一致」
- 条件値：<%REQUEST\_PARAMETER(regionNameParameter)%>

上 へ	下 へ	テーブル/ビュー/リソース	カラム/フィールド	抽出方法	条件値	削除
↑	↓	sample_population	year	以上	<%REQUEST_PARAMETER(minimumYearParameter)%>	✖
↑	↓	sample_population	year	以下	<%REQUEST_PARAMETER(maxYearParameter)%>	✖
↑	↓	sample_region	name	部分一致	<%REQUEST_PARAMETER(regionNameParameter)%>	✖

図：「クエリ編集」画面 - 抽出条件追加設定

9. 「クエリ編集」画面上部の「保存」ボタンをクリックします。

10. 「確認」ダイアログでクエリ名を「query\_issue\_32032」に変更して「決定」をクリックします。

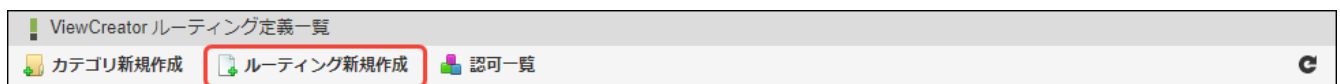


図：「クエリ編集」画面 - クエリ保存

## ルーティング定義の作成

次に、ルーティング定義を作成します。

1. 「サイトマップ」→「ViewCreator」→「ルーティング定義一覧」から、「ルーティング定義一覧」画面を表示します。
2. 追加先のルーティングカテゴリをルーティングツリーから選択し、「ルーティング定義一覧」画面左上の「ルーティング新規作成」をクリックします。  
ルーティングカテゴリ未作成の場合は、「[ルーティングカテゴリの作成](#)」の説明を参考にルーティングカテゴリを作成してください。



図：「ルーティング定義一覧」画面 - 「ルーティング新規作成」

3. ルーティング情報の各項目を以下のとおりに設定します。
  - ルーティングID : `routing_issue_32032`
  - ルーティングURL : `viewcreator/sample/routes/api/get`
  - 対象クエリ : 「[実行するクエリの作成](#)」で作成したクエリ定義
  - メソッド : 「GET」
  - 認証方式 : 「IMAuthentication」
  - レスポンス種別 : 「JSONに変換して返却」
  - ルーティング名 : `SampleRouting(issue32032)`



図：「ルーティング定義一覧」画面 - ルーティング情報の入力

4. 「登録」をクリックしてルーティング定義を登録します。
5. 「認可URI」の「認可設定」アイコンをクリックします。

図：「認可URI」 - 「認可設定」アイコン

## 6. 「認可設定」画面で、作成したルーティングに対する認可設定を行います。

本作成例では、対象者条件の「認証済みユーザ」に対して「認可」を設定します。



図：「認可設定」画面 - 「認証済みユーザ」へ認可設定

## 画面の作成

続いて、IM-BloomMakerを使用して画面を作成します。

IM-BloomMakerで作成する画面定義（コンテンツおよびルーティング）については、以下の完成サンプルをダウンロードしてご活用ください。

IM-BloomMakerインポートデータ：[viewcreator\\_routing\\_bm\\_sample\\_issue\\_32032](#)

## i コラム

本作成例では、IM-BloomMakerによる画面の作成手順については説明しません。

画面の作成については、「[IM-BloomMaker ユーザ操作ガイド](#)」や「[IM-BloomMaker チュートリアルガイド](#)」および [intra-mart Developer Site](#) の [CookBook](#) をご活用ください。

1. 「サイトマップ」→「BloomMaker」→「インポート」をクリックし、IM-BloomMakerのインポート機能を利用して完成サンプルをインポートします。

## i コラム

インポート手順については、「[IM-BloomMaker ユーザ操作ガイド](#)」の「[定義ファイルをインポートする](#)」を参照してください。

2. 完成サンプルに含まれる以下のルーティング定義に対して、「[ルーティング定義の作成](#)」の認可設定と同じく「認証済みユーザ」の「認可」を設定します。
  - ルーティングID：`routing_issue_32032`
  - ルーティング名：`ViewCreator SampleRouting(issue32032)`

## i コラム

ルーティング定義の認可設定については、「[IM-BloomMaker チュートリアルガイド](#)」の「[ルーティングの認可を設定する](#)」を参照してください。

## ! 注意

ViewCreatorのルーティング定義とIM-BloomMakerのルーティング定義の認可対象を揃えてください。

認可にずれがある場合、ユーザによっては画面の表示のみ可能（クエリの実行が不可能）な状態が発生する可能性があります。

3. 認証済みユーザで「<http://localhost:8080/imart/viewcreator/sample/routes/view/list>」にアクセスし、本作成例のアプリケーション画面が表示されることを確認します。

**i** コラム

ベースURLである以下の部分は環境に合わせて適宜変更してください。

http://localhost:8080/imart

## IM-BloomMakerとの連携

本作成例における、IM-BloomMakerとREST APIの連携について説明します。

- REST APIの実行

IM-BloomMakerのアクションには、指定したURLにリクエストを送信するアクションアイテムが用意されています。

このアクションアイテムを利用することで、ViewCreatorで作成したREST APIを実行できます。

定数 VIEWCREATOR\_ROUTING\_URL には、「ルーティング定義の作成」で設定したルーティングURLを指定します。

```
{
  "VIEWCREATOR_ROUTING_URL": "viewcreator/sample/routes/api/get"
}
```

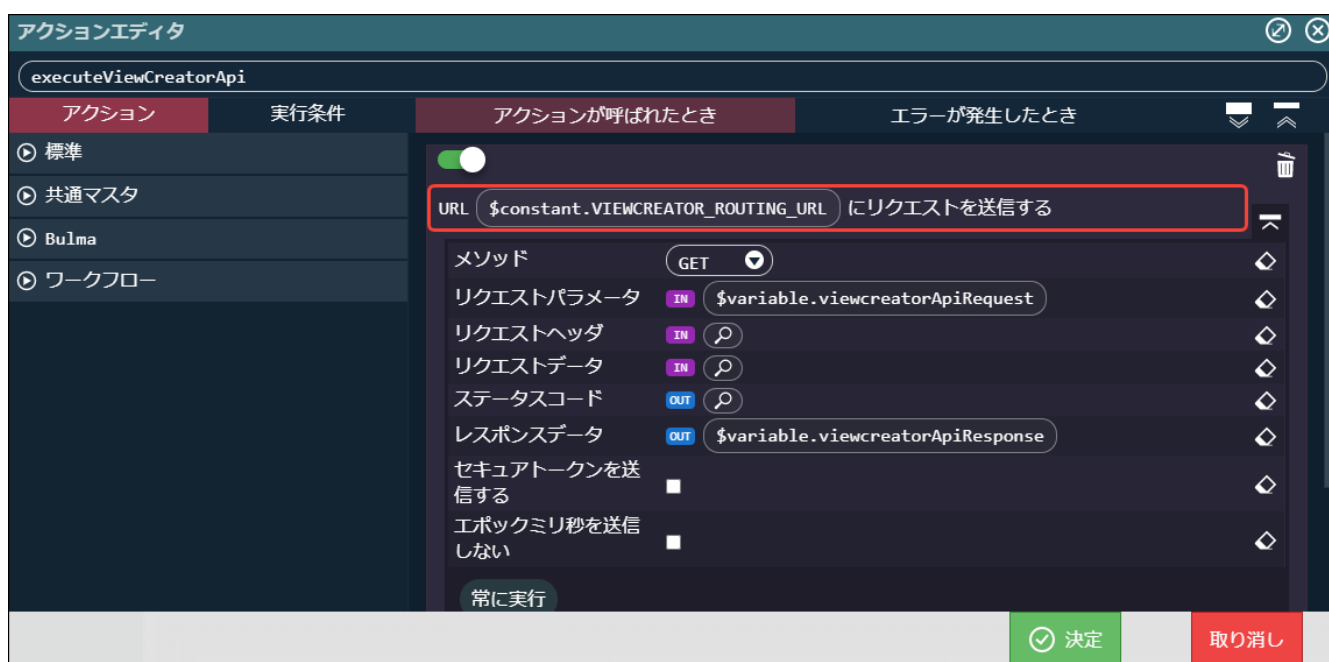


図 : executeViewCreatorApi アクション

- REST APIのリクエストパラメータの設定

IM-BloomMakerのアクションアイテムには、リクエストパラメータとして渡す変数を設定する機能が用意されています。

リクエストパラメータを格納する変数を定義することで、REST APIのリクエストパラメータを設定できます。

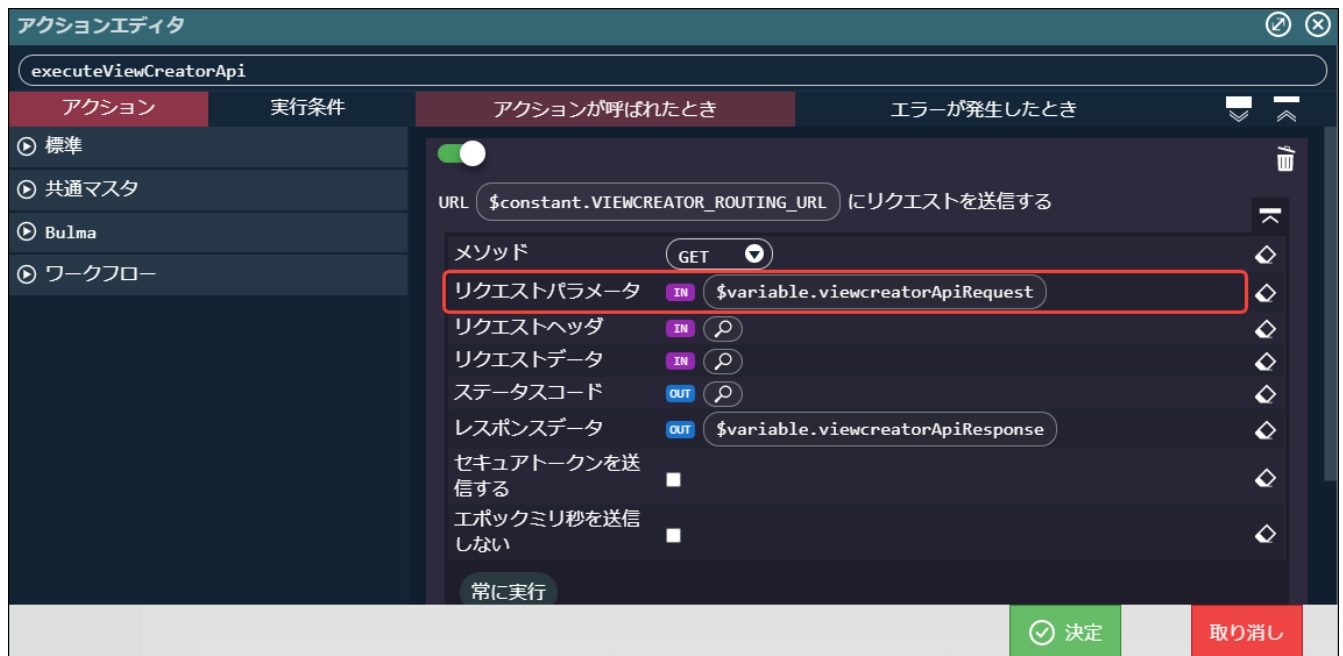


図 : executeViewCreatorApi アクション - リクエストパラメータ

本作成例では、リクエストパラメータを設定する変数を以下のように定義しています。

キー名	説明
<code>viewcreatorApiRequest.limit</code>	最大取得レコード件数を設定する <code>limit</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.offset</code>	レコードの取得開始位置を設定する <code>offset</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.columns</code>	取得対象とするカラムを設定する <code>columns</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.sort</code>	第1ソートキーを設定する <code>sort</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.order</code>	第1ソートキーに対するソート方向を設定する <code>order</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.secondSort</code>	第2ソートキーを設定する <code>secondSort</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.secondOrder</code>	第2ソートキーに対するソート方向を設定する <code>secondOrder</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.thirdSort</code>	第3ソートキーを設定する <code>thirdSort</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.thirdOrder</code>	第3ソートキーに対するソート方向を設定する <code>thirdOrder</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.responseKeyType</code>	レスポンスデータのキー項目を設定する <code>responseKeyType</code> パラメータの値を保持します。
<code>viewcreatorApiRequest.minimumYearParameter</code>	クエリの抽出条件（動的パラメータ <code>&lt;%REQUEST_PARAMETER(minimumYearParameter)%&gt;</code> ）の値を保持します。
<code>viewcreatorApiRequest.maxYearParameter</code>	クエリの抽出条件（動的パラメータ <code>&lt;%REQUEST_PARAMETER(maxYearParameter)%&gt;</code> ）の値を保持します。
<code>viewcreatorApiRequest.regionNameParameter</code>	クエリの抽出条件（動的パラメータ <code>&lt;%REQUEST_PARAMETER(regionNameParameter)%&gt;</code> ）の値を保持します。

#### ■ REST APIの実行結果の取得

IM-BloomMakerのアクションアイテムには、レスポンスデータを格納する変数を設定する機能が用意されています。レスポンスデータを格納する変数を定義することで、REST APIの実行結果を変数を通して取得できます。

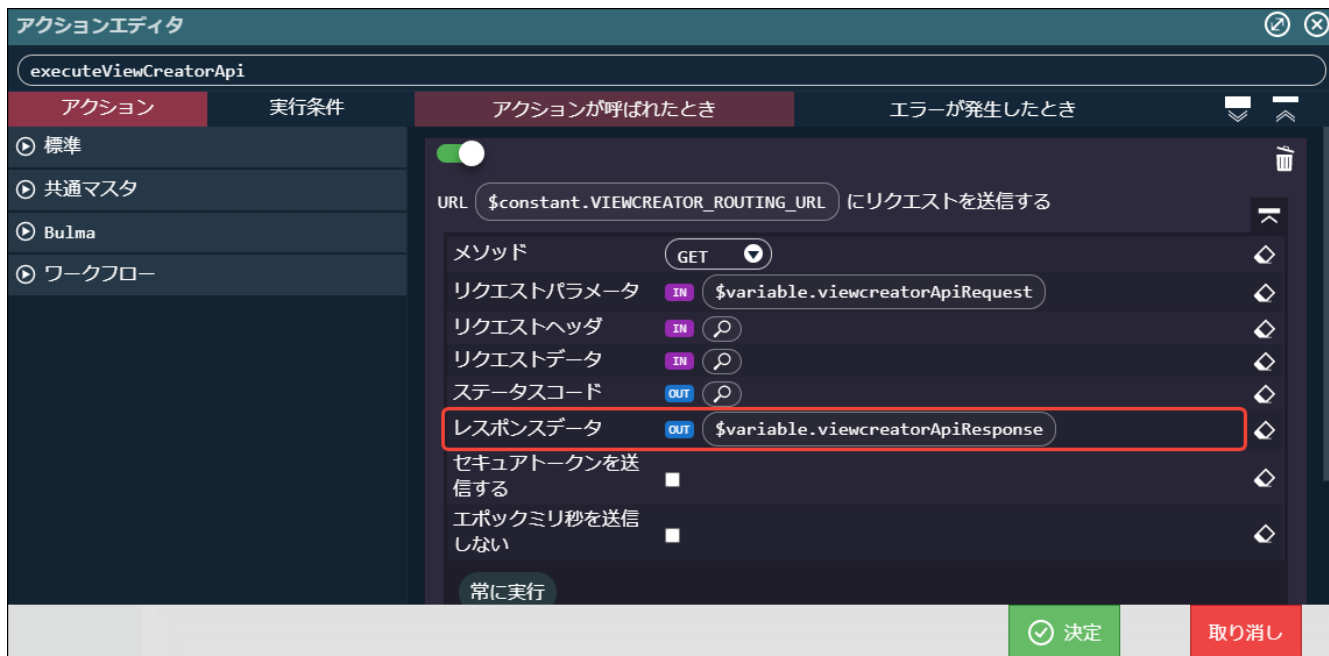


図 : executeViewCreatorApi アクション - レスポンスデータ

本作成例では、レスポンスデータを格納する変数を以下のように定義しています。

キー名	説明
<code>viewcreatorApiResponse.data</code>	REST APIのレスポンスボディの <code>data</code> パラメータの格納先です。 取得対象の各カラムの値を格納する変数で構成された配列型の変数です。 変数の名称は、 <code>responseKeyType</code> パラメータに「CAPTION」を設定しているため、カラムの論理名 ( <code>year</code> , <code>regionName</code> , <code>prefectureName</code> , <code>age</code> , <code>area</code> , <code>population</code> ) を使用します。
<code>viewcreatorApiResponse.totalCount</code>	REST APIのレスポンスボディの <code>totalCount</code> パラメータの格納先です。

■ REST APIの実行結果と画面の連携

REST APIの実行結果を格納する変数と画面上に配置したエレメントを関連付けることで、実行結果と画面を連携できます。本作成例では、画面上に配置したエレメントと変数を以下のように関連付けています。

エレメント	説明
テーブルコンテナ (繰り返し)	REST APIの実行結果の一覧 ( <code>data</code> パラメータ) と関連付けるために、 <code>list</code> プロパティに変数 <code>viewcreatorApiResponse.data</code> を設定します。 また、各テーブルセルに配置した「ラベル」エレメントにREST APIの実行結果を表示するために、 <code>textContent</code> プロパティに各カラムデータの格納先変数 ( <code>year</code> , <code>regionName</code> , <code>prefectureName</code> , <code>age</code> , <code>area</code> , <code>population</code> ) を設定します。
ページネーション	REST APIの実行結果の総件数 ( <code>totalCount</code> パラメータ) と関連付けるために、「ページネーションリスト」エレメントの <code>total</code> プロパティに <code>viewcreatorApiResponse.totalCount</code> を設定します。

