Copyright © 2012 NTT DATA INTRAMART CORPORATION

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
- 5. 注意事項
  - 5.1. データ参照表示時のリソースの使用量増加について

改訂情報

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 ▪ 「*対応するフィールドの型*」を追加 ▪ 「*外部データソースの利用*」を追加
2013-07-01	第3版 下記を追加・変更しました ▪ 「*リソースの設定項目*」を追加 ▪ 「*キー名の変更*」を追加 ▪ 「*計算式カラムの設定*」を追加 ▪ 「*リスト集計におけるCSV出力の設定*」を追加
2013-10-01	第4版 下記を追加・変更しました ▪ 「*対応するフィールドの型*」を追加 ▪ 「*サマリ集計の設定*」を追加 ▪ 「*表示設定*」を追加 ▪ 「*リスト集計からグラフ集計を作成*」を追加
2014-01-01	第5版 下記を追加・変更しました ▪ 「*アイコン画像の表示*」を追加 ▪ 「*検索設定について*」に設定可能な動的パラメータについての記載を追加 ▪ 「*外部ページへの連携（リスト/サマリ集計のみ）*」にカラム単位でのURL設定ついての記載を追加 ▪ 「*外部ページへの連携（リスト/サマリ集計のみ）*」にターゲットウィンドウについての記載を追加 ▪ 「*外部ページへの連携（リスト/サマリ集計のみ）*」にPathVariablesについての記載を追加 ▪ 「*国際化とヘッダ・フッタ*」に カラムのラベルを国際化対応についての記載を追加 ▪ 「*インポート・エクスポート*」にインポート・エクスポートに関する注意事項を追加 ▪ 「*クエリの作成*」にインポート・エクスポートに関する注意事項を追加
2014-04-01	第6版 下記を追加・変更しました ▪ 「*利用可能な関数一覧*」にFORMAT関数へのロケールの指定についての記載を追加 ▪ 「*参照権の設定*」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 ▪ 「*国際化とヘッダ・フッタ*」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 ▪ 「*計算式カラムの設定*」に計算式入力ダイアログについての記載を追加 ▪ 「 *作成したデータ参照の閲覧*」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 ▪ 「 *動的パラメータ*」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「<%REQUEST_PARAMETER%>タグ」を追加
2014-08-01	第7版 下記を追加・変更しました ▪ 新しいデータ型（真偽値型）を対応 ▪ 「*複数の表示条件設定（ヒートマップの作成）*」を追加 ▪ 「*CSV出力ジョブの登録*」を追加 ▪ 「*ソースコード出力*」を追加 ▪ 「*帳票出力の設定*」を追加 ▪ 「*null値に対する表示文字列の設定*」を追加 ▪ 「*応用設定*」-「グラフ集計用モジュールの切り替え」を追加 ▪ 「*利用可能な関数一覧*」に関数を追加

変更年月日	変更内容
2014-12-01	第8版 下記を追加・変更しました ▪ 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 ▪ 「*グラフ集計の作成*」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 ▪ 「*リスト集計におけるCSV出力の設定*」に設定可能項目を追加 ▪ 「検索の条件値入力にコンボボックスを利用する」を追加 ▪ 「*表示設定*」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 ▪ 「*グラフ集計で「凡例」の表示方法を変更する*」を追加 ▪ 「*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*」を追加
2015-04-01	第9版 下記を追加・変更しました ▪ 「*インポート・エクスポート*」に IM-FormaDesigner for Accel Platform についての注意事項を追加 ▪ 「*計算式カラムの設定*」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
2015-08-01	第10版 下記を追加・変更しました ▪ 「*ViewCreator全体の権限設定*」に認可リソース「管理者」についての記載を追加
2015-12-01	第11版 下記を追加・変更しました ▪ 「*クロス集計*」の画面インタフェースと機能を一新 ▪ 「*応用設定*」-「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 ▪ 「*作成したデータ参照の閲覧*」に「CSVファイルのヘッダ文字列」に関するコラムを追加 ▪ 「*リスト集計の作成*」のソート順に関するコラムに記載を追加
2016-04-01	第12版 下記を追加・変更しました ▪ 「*リスト集計におけるCSV出力の設定*」にBOMに付与設定についての記載を追加 ▪ 「*動的パラメータ*」の<%REQUEST_PARAMETER(KEY_NAME)%>の説明、および、使用例に関するコラムに記載を追加 ▪ 「*検索設定について*」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 ▪ 「*ソースコード出力*」に出力されたソースコードの利用手順を追加
2016-08-01	第13版 下記を追加・変更しました ▪ 「*計算式カラムの設定*」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 ▪ 「*グラフ集計の作成*」-「*JFreeChart*」を追加 ▪ 「*グラフ集計の作成*」-「*Highcharts*」を追加 ▪ 「*応用設定*」-「グラフ集計用モジュールの切り替え」を削除
2016-12-01	第14版 下記を追加・変更しました。 ▪ 「*対応するフィールドの型* 」にSAP HANAに関する記載を追加、DB2に関する記載を削除 ▪ 「*クロス集計* 」にクロス集計のカラム設定に注意事項を追加
2017-08-01	第15版 下記を追加・変更しました。 ▪ 「*クエリの作成* 」の説明を修正 ▪ 「*クエリの作成* 」に「*クエリの作成（2017 Spring(Portland) 以前）* 」に関するコラムを追加 ▪ 「*抽出条件の設定* 」の説明を修正 ▪ 「*外部データソースの利用* 」の説明を修正 ▪ 「*REST APIを参照する* 」を追加

変更年月日	変更内容
2017-12-01	第16版 下記を追加・変更しました。 ▪ 「*対応するフィールドの型*」にNCLOBを追加 ▪ 「*リスト集計におけるCSV出力の設定* 」にfetch-sizeについての記載を追加 ▪ 「*外部データソースの利用*」-「*ロジックフローの利用* 」を追加
2018-02-01	第17版 下記を追加・変更しました。 ▪ 「*設定可能なパラメータ*」のdest_storage_pathの説明を修正
2018-04-01	第18版 下記を追加・変更しました。 ▪ 「*設定可能なパラメータ*」にdest_storage_file_nameの説明を追加 ▪ 「*生成されたソースコードを動作させるための手順*」にintra-mart Accel Platform 2018 Spring(Skylark)以降用の説明を追加
2018-08-01	第19版 下記を追加・変更しました。 ▪ 「*検索の条件値入力に選択型コンポーネントを利用する*」を追加
2019-04-01	第20版 下記を追加・変更しました ▪ 「*リスト集計におけるCSV出力の設定*」のinclude-binaryの説明を修正 ▪ 「*データ型と送信データについて*」を追加 ▪ 「*検索設定について*」に表示状態の保持についての説明を追加 ▪ 「*クロス集計*」に「*小計行の表示*」を追加 ▪ 「*クロス集計*」に「*ソートについて*」を追加 ▪ 「*クロス集計*」に「*AND検索とOR検索（各項目を結合する演算子）について*」を追加
2019-08-01	第21版 下記を追加・変更しました ▪ 「*利用可能な関数一覧*」のCOLUMNVALUES、LOOKUP、COUNTIF、IF、SUBSTITUTE関数の説明を追加 ▪ 「*リスト集計におけるCSV出力の設定*」のdate-format、decimal-formatに説明を追加 ▪ 「*リスト集計におけるCSV出力の設定*」に「*フォーマット適用の優先順について*」を追加 ▪ 「*データ参照の作成*」の「*リスト集計の作成*」に項目説明を追加 ▪ 「*データ参照の作成*」の「*検索設定について*」にダウンロード時の説明を追加 ▪ 「*JFreeChart*」のURLを最新化 ▪ 「*Highcharts*」のURLを最新化
2019-12-01	第22版 下記を追加・変更しました ▪ 「*Highcharts*」のグラフ画像をHighchartsのバージョンアップに伴い差し替え ▪ 「*Highcharts*」の「*グラフ設定*」にグラフ表示時の注意事項を追加 ▪ 「*利用可能な関数一覧*」にROWNUMBER関数を追加

変更年月日	変更内容
2020-04-01	第23版 下記を追加・変更しました ▪ 「*Highcharts*」のグラフ種類に「*エリアチャート*」を追加 ▪ 「*Highcharts*」のカラム設定にキャプション軸の説明を追加 ▪ 「*Highcharts*」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 ▪ 「*クエリ一覧画面*」のクエリ名項目にクエリ名のアイコンの説明を追加 ▪ 「*データ参照の種類の選択*」のクエリ一覧にデータ参照作成のアイコンの説明を追加 ▪ 「*作成したデータ参照の閲覧*」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 ▪ 「*ViewCreator について*」にSQLビルダの説明を追加 ▪ 「*ViewCreator全体の権限設定*」にSQLビルダに関する認可の説明を追加 ▪ 「*クエリの作成*」からクエリ編集画面の詳細の説明を「*クエリビルダによるクエリの作成*」と「*SQLビルダによるクエリの作成*」に変更 ▪ 「*外部データソースの利用*」にSQLビルダにおいての注意事項を追加 ▪ 「*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*」にSQLビルダにおいての注意事項を追加 ▪ 「*物理Viewの作成と表示*」にSQLビルダにおいての注意事項を追加 ▪ 「*クエリの作成*」に「*クエリ一覧に表示されるクエリについて*」を追加
2020-08-01	第24版 下記を変更しました ▪ 「*計算式カラムの設定*」のヘッダ・フッタの計算式設定画像を差し替え ▪ 「*リスト集計の作成*」のデータ参照編集画面の画像を差し替え ▪ 「*リスト集計の作成*」にタイムゾーン設定の説明を追加 ▪ 「*利用可能な関数一覧*」にTIMEZONE関数を追加
2020-12-01	第25版 下記を追加・変更しました ▪ 「*ViewCreator全体の権限設定*」に認可リソース「ルーティングの作成」についての記載を追加 ▪ 「*インポート・エクスポート*」のインポート・エクスポートに関する注意事項を修正 ▪ 「*REST APIを参照する*」のREST APIの説明および画像を追加 ▪ 「*ルーティングの作成*」を追加
2021-04-01	第26版 下記を追加・変更しました ▪ 「*ViewCreator全体の権限設定*」の認可リソース「インポート・エクスポート 」に管理対象機能を追加 ▪ 「*インポート・エクスポート*」に追加機能の説明を追加 ▪ 「*ルーティングの作成*」の「*ルーティングの実行確認*」の説明を修正 ▪ 「*ルーティングの作成*」に「*REST APIの仕様確認*」の説明を追加 ▪ 「*外部データソースの利用*」の「*ロジックフローを検索する*」にフローカテゴリと検索結果の表示の説明を追加
2021-08-01	第27版 下記を追加・変更しました ▪ 「*検索設定について*」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
2021-12-01	第28版 下記を追加・変更しました ▪ 「*インポート・エクスポート*」の「*クエリとデータ参照のインポート/エクスポート*」に非推奨である注意事項を追加 ▪ 「*インポート・エクスポート*」の「*インポート*」の説明および画像を修正 ▪ 「*インポート・エクスポート*」の「*エクスポート*」の説明および画像を修正
2022-06-01	第29版 下記を追加・変更しました ▪ 「*データ参照の作成*」の「*クロス集計の編集画面*」にクロス集計のメモリ使用量に関する設定について説明を追加 ▪ 「*利用可能な関数一覧*」に SWITCHMULTI関数の説明を追加

変更年月日	変更内容
2023-04-01	第30版 下記を追加・変更しました ▪ 「*データ参照の作成*」に「*参照画面の操作状態について*」を追加 ▪ 「*データ参照の作成*」に「*検索タイプについて*」を追加 ▪ 「*注意事項*」に「*データ参照表示時のリソースの使用量増加について*」を追加
2023-10-01	第31版 下記を追加・変更しました ▪ 「*データ参照の作成*」に「*入力制限について*」を追加 ▪ 「*外部データソースの利用*」の「*リソースの設定項目*」にリソースパスの説明と注意事項を追加 ▪ 「*帳票出力の設定*」にブラウザ印刷設定の説明を追加
2024-04-01	第32版 下記を追加しました ▪ 「*動的パラメータを利用可能な個所*」を追加 ▪ 「*ルーティングの作成*」に「*ルーティングの OAuth スコープ設定*」を追加 ▪ 「*外部データソースの利用*」-「*ロジックフローの利用*」に注意事項を追加

ViewCreator について

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。

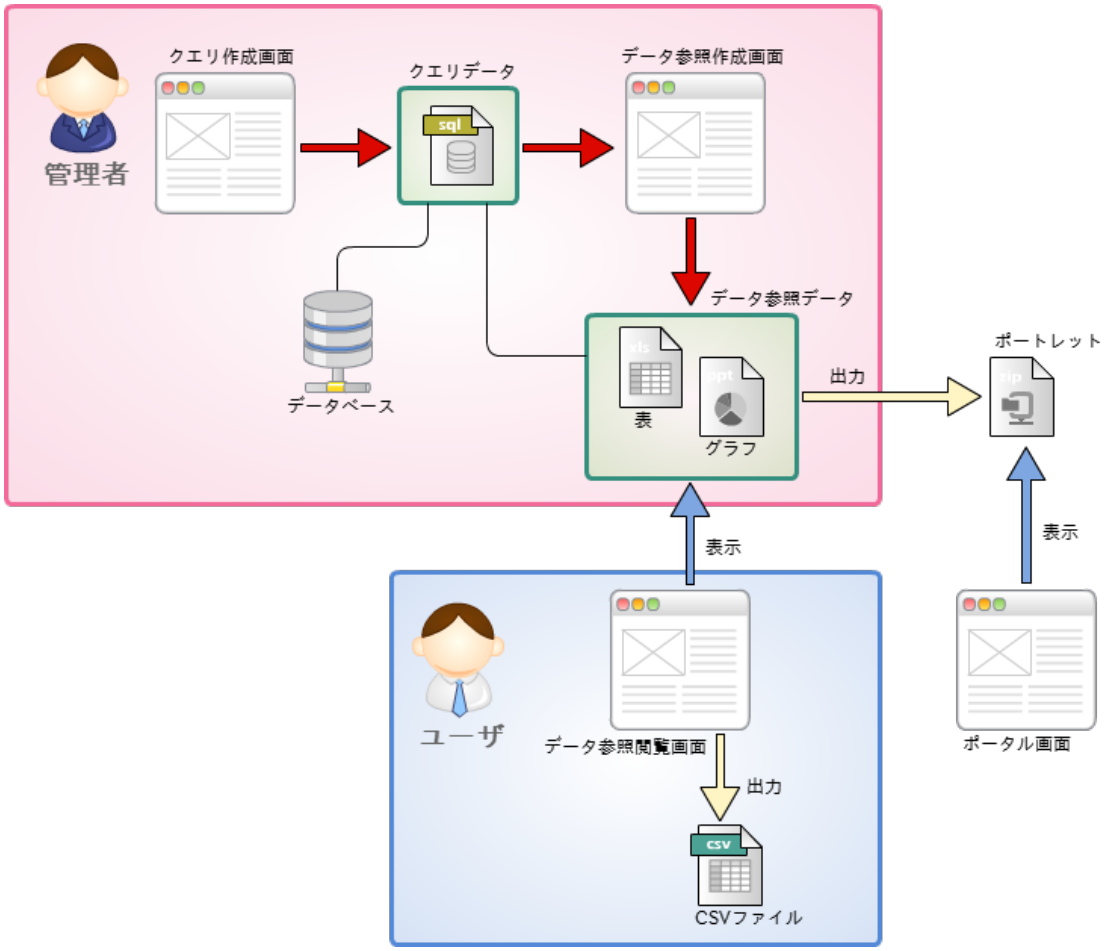
- **クエリ**

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。



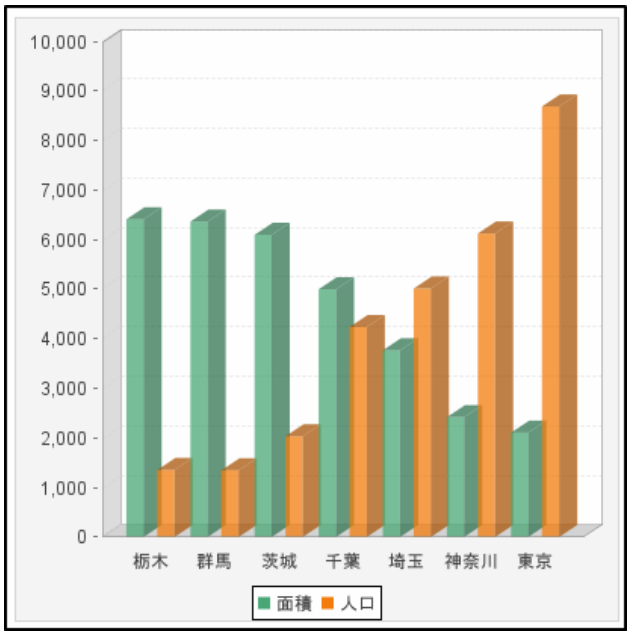
図：ViewCreator 概要図

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。

1

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。

基本設定

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

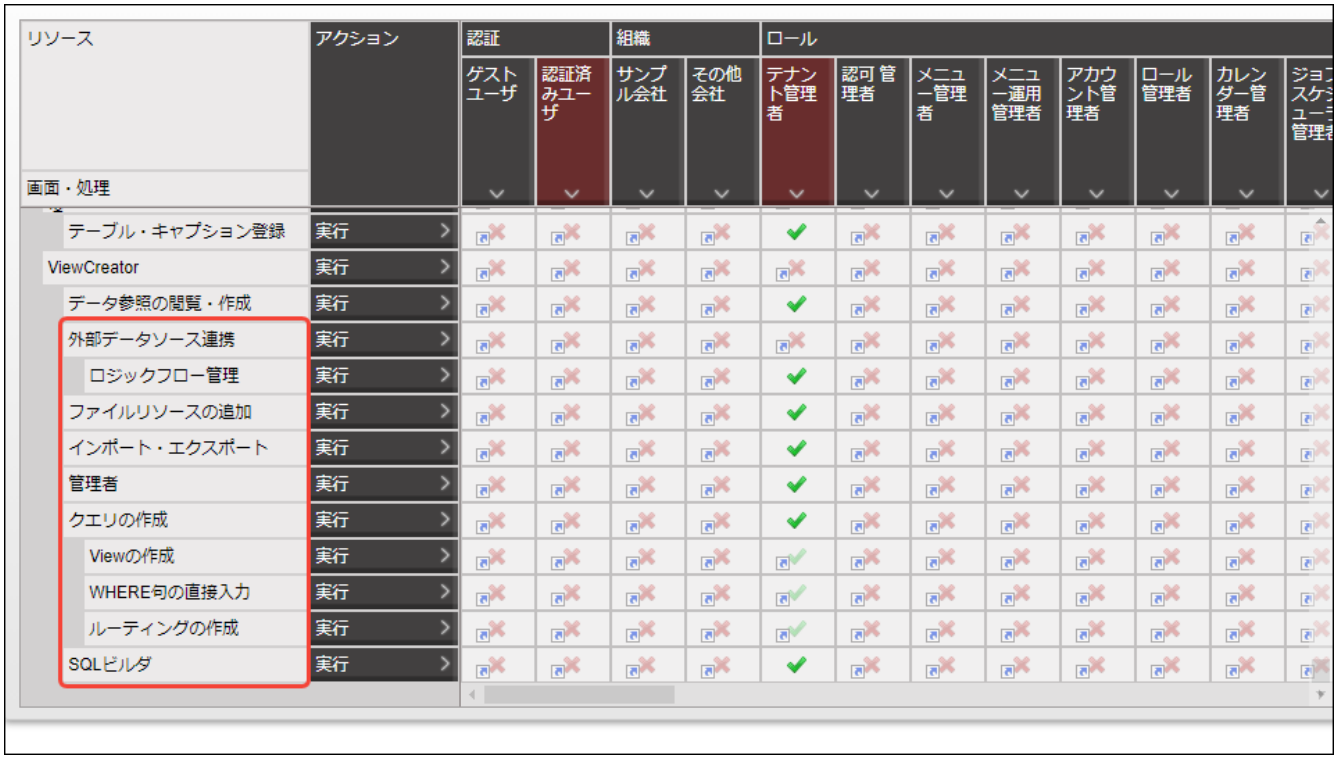
まず最初に一般ユーザがViewCreatorのメニューを使えるように設定します。

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。



リソース名	説明
データ参照の閲覧・作成	データ参照一覧へのアクセスを可能にします。 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。 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。 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。 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。
ファイルリソースの追加	外部ファイルリソースの追加・削除を可能にします。 一般的には管理者ユーザのみを使用可能に設定します。

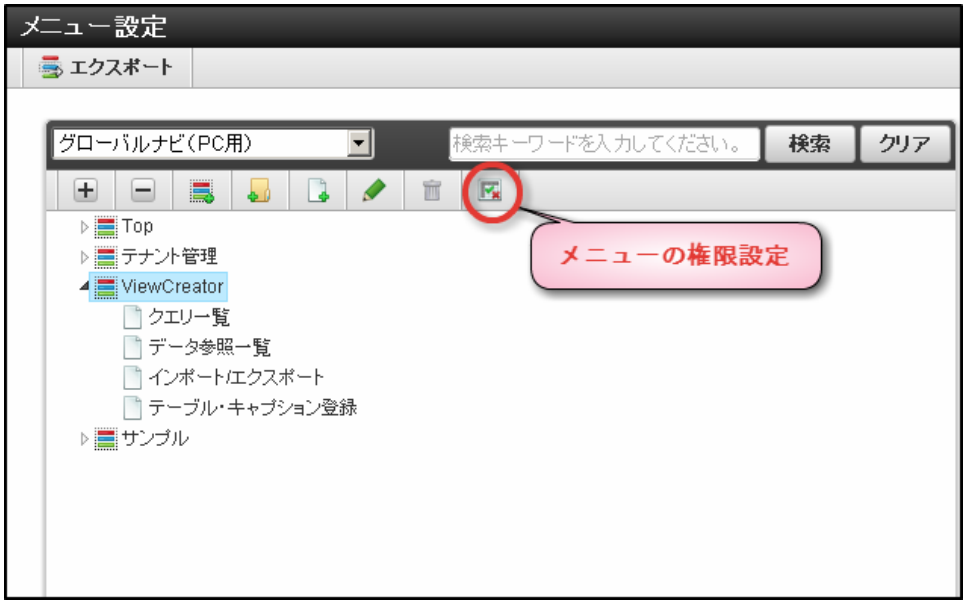
リソース名	説明
インポート・エクスポート	ViewCreatorの設定データを一括でインポート/エクスポートする機能へのアクセスを可能にします。 一括でインポート/エクスポートする機能として以下の機能が管理されます。 ・ クエリとデータ参照のインポート/エクスポート ・ インポート ・ エクスポート 個別でインポート/エクスポートする機能は管理外です。 一般的には管理者ユーザのみを使用可能に設定します。
管理者	ViewCreatorを管理者として利用できるようにします。 具体的には、管理者ユーザは登録されている全てのクエリやデータ参照に対して変更や削除が可能です。 ただし、クエリ一覧画面や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「*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。
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「*ルーティングの作成*」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「*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。

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。

> **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。

> **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

- クエリ一覧画面
  - クエリ一覧に表示されるクエリについて
- クエリ編集画面

クエリ一覧画面

「サイトマップ」→「ViewCreator」→「クエリ一覧」をクリックします。

図：クエリ一覧画面

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「*データ参照の作成*」を参照してください。
クエリ名	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。 *クエリビルダ* で作成したクエリです。 SQL *SQLビルダ* 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。

クエリ一覧に表示されるクエリについて

クエリ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

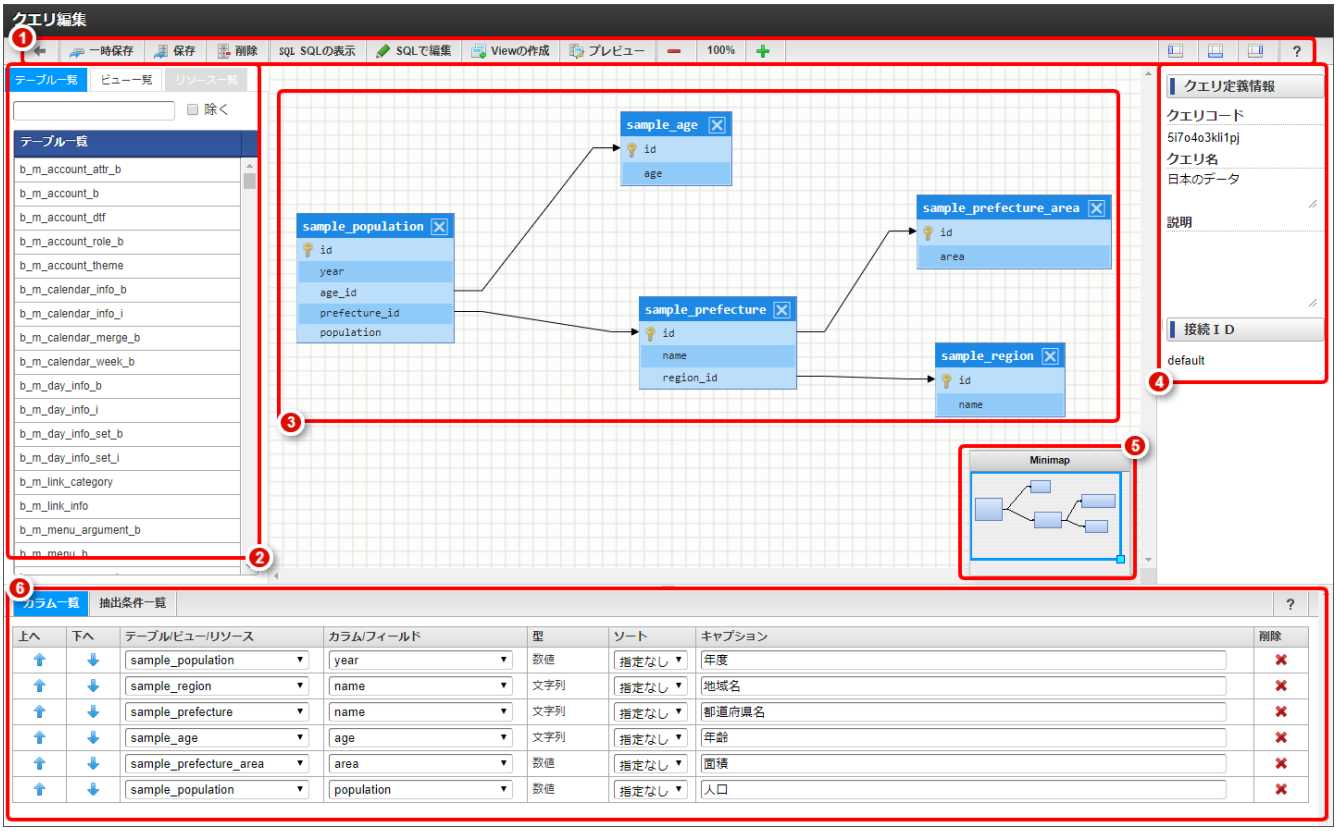
クエリビルダではドラッグ&ドロップなどを用いたGUI操作でクエリの作成が行えます。

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

1. 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「*物理Viewの作成と表示*」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。

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。

コラム

カ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「カラム/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。
カ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。

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。

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ド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。

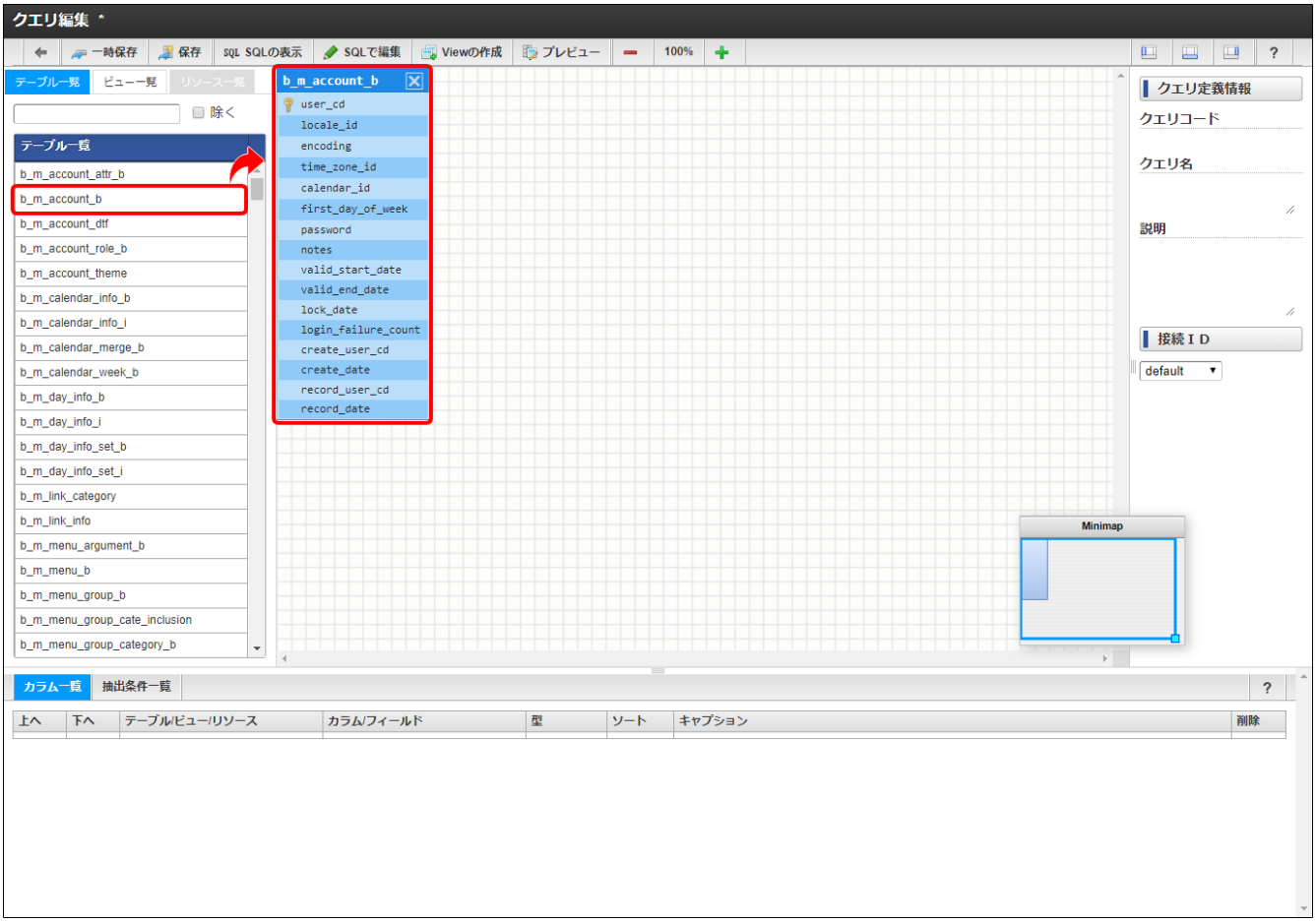
5. デザイナ部分の全体図を表します。
   青い枠に囲まれた部分が現在の描写範囲で、ドラッグすることで描写範囲を変更できます。

6. クエリのカ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カラム一覧および抽出条件一覧を表示します。

抽出条件の詳細は「*抽出条件の設定*」を参照ください。

テーブルの追加

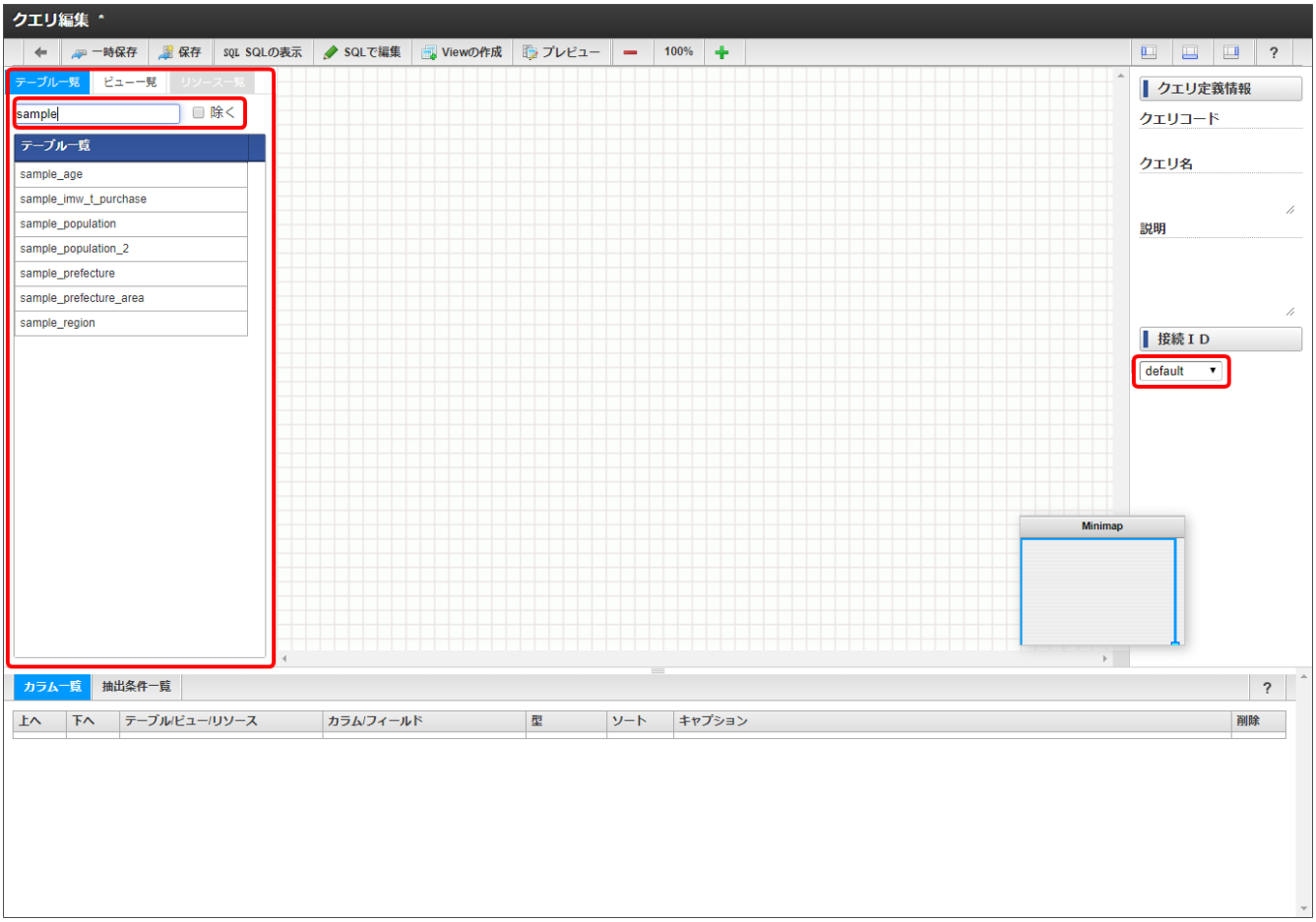
「テーブル一覧」から追加するテーブルをダブルクリックすることで、デザイナにテーブルウィンドウが表示されます。
「カギ」アイコンは主キーを表します。



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。



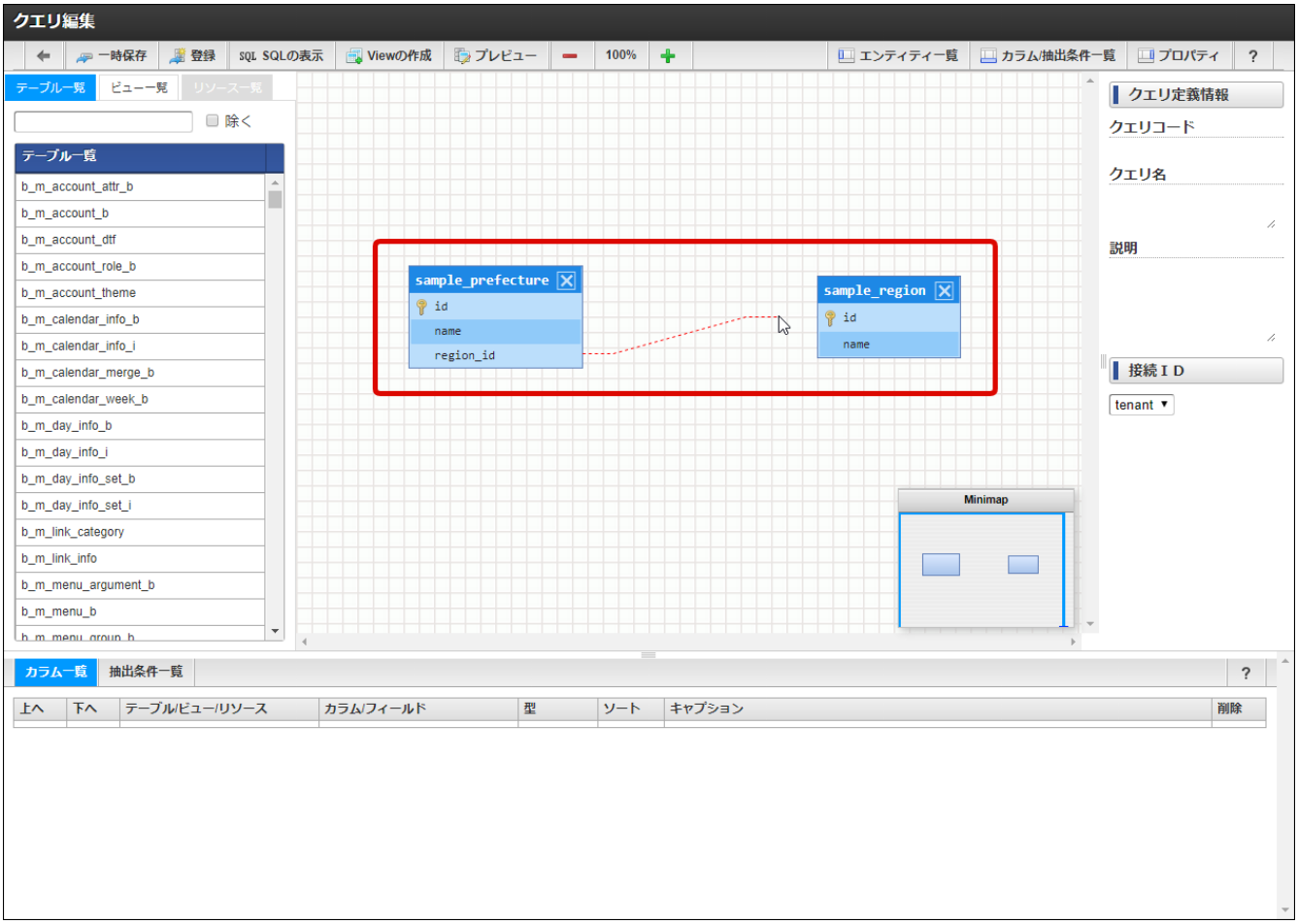
図：クエリ編集画面 - テーブル一覧

> **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。

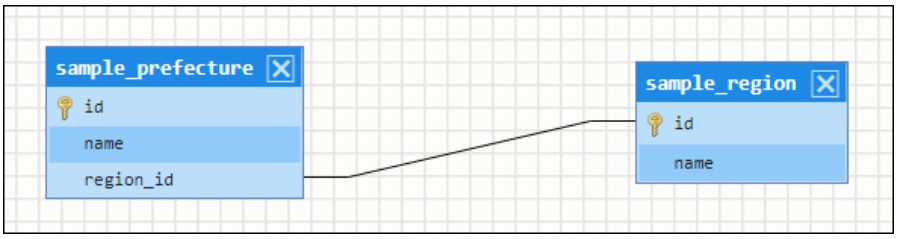
結合条件の作成

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

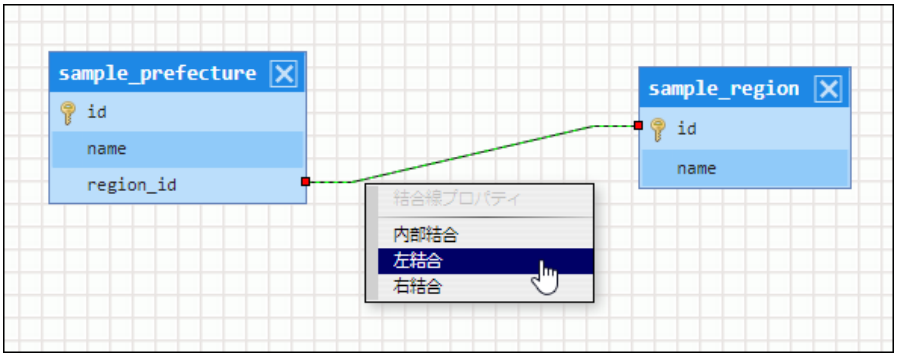
結合条件（結合線）が描画されます。



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。

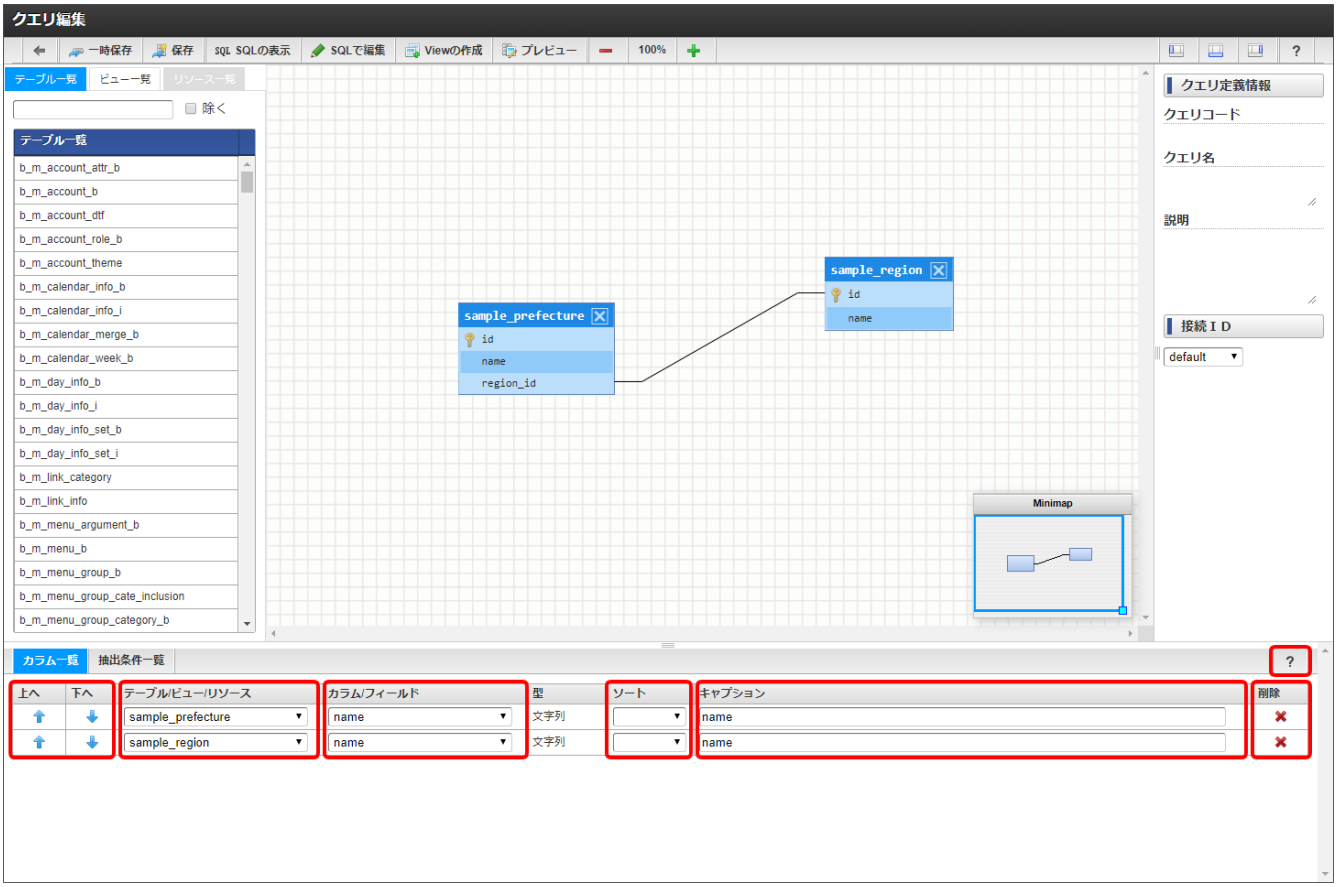


図：クエリ編集画面 - 結合条件変更

> **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。

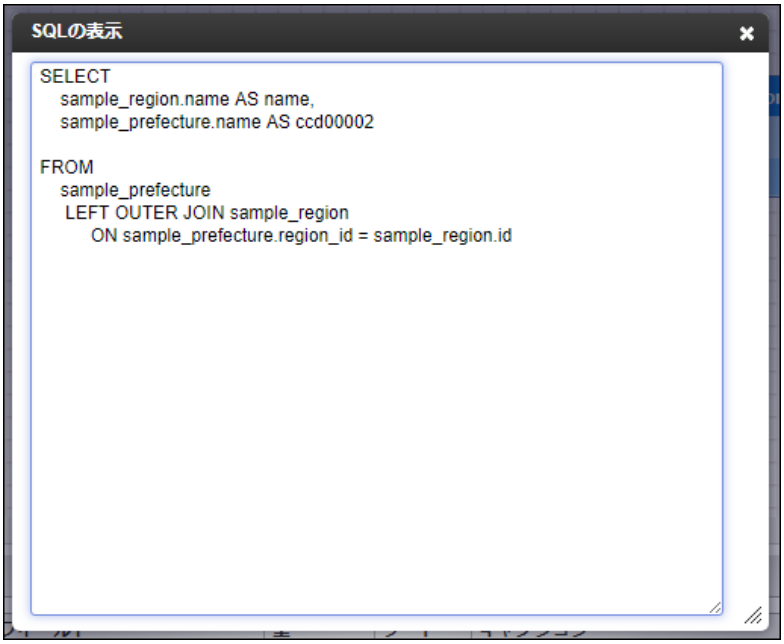


図：クエリ編集画面 - カラム追加

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。



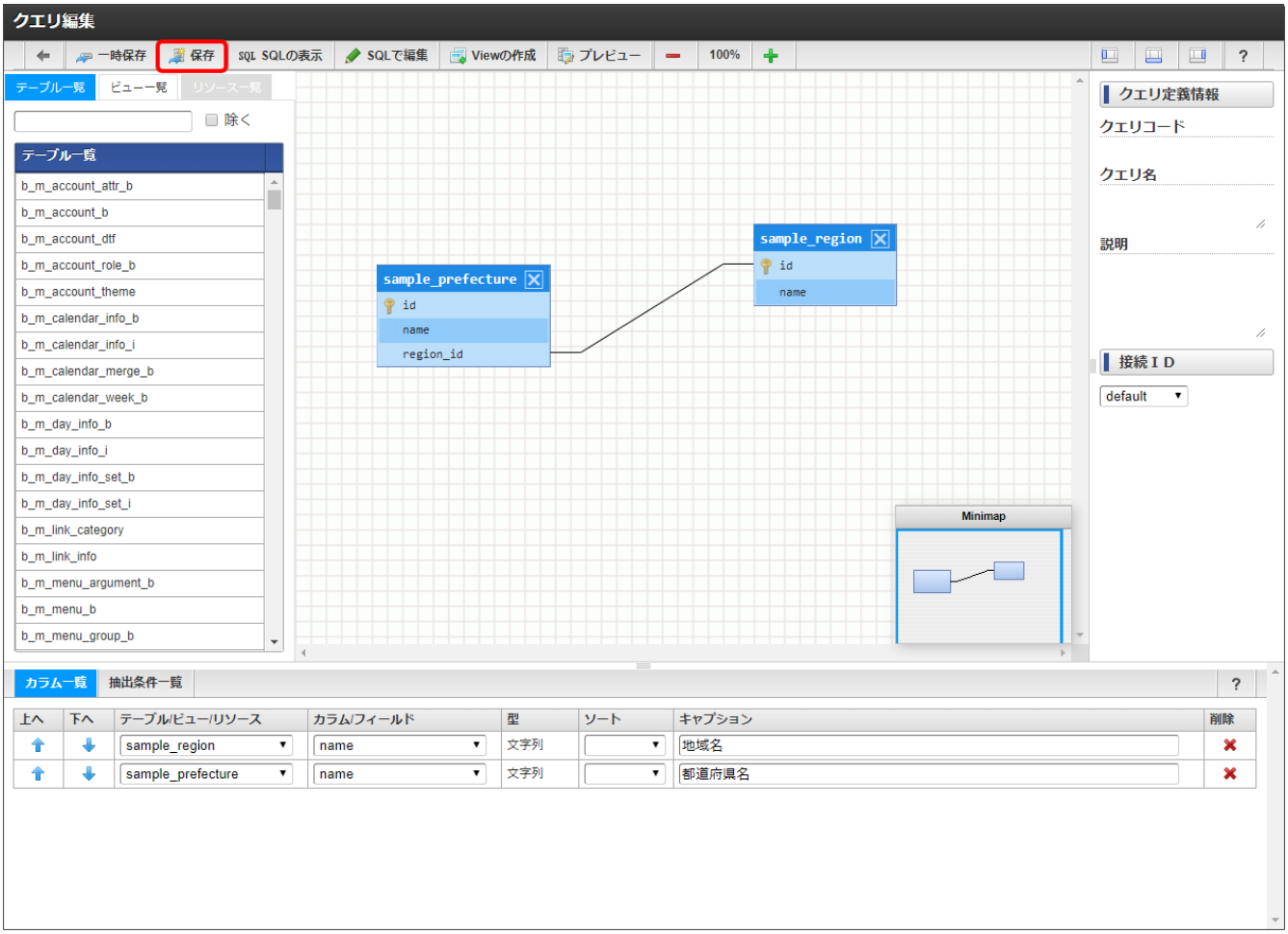
図：クエリ編集画面 - SQLの表示

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カラムは、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ビルダによるクエリの作成

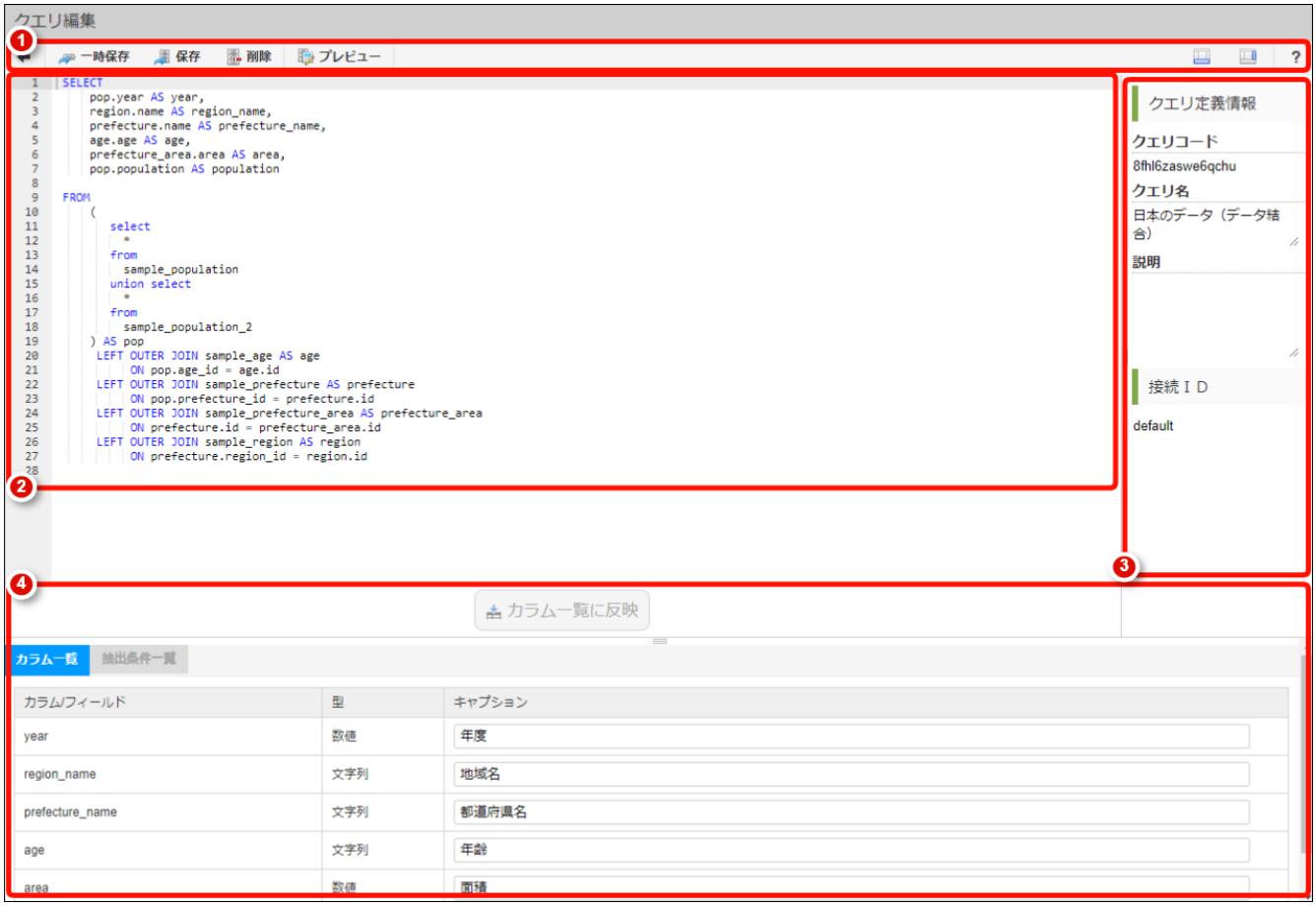
SQLビルダではSQLを直接入力してクエリの作成が行えます。

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。



図：クエリ編集画面 - 初期表示

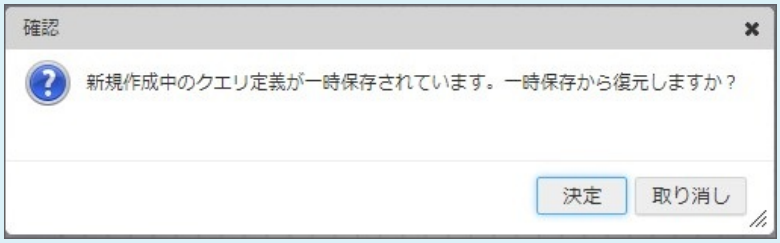
1. クエリ編集に対する基本的な操作を提供するヘッダです。
   表示項目は以下の通りです。

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と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と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。

2. 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。

3. 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。

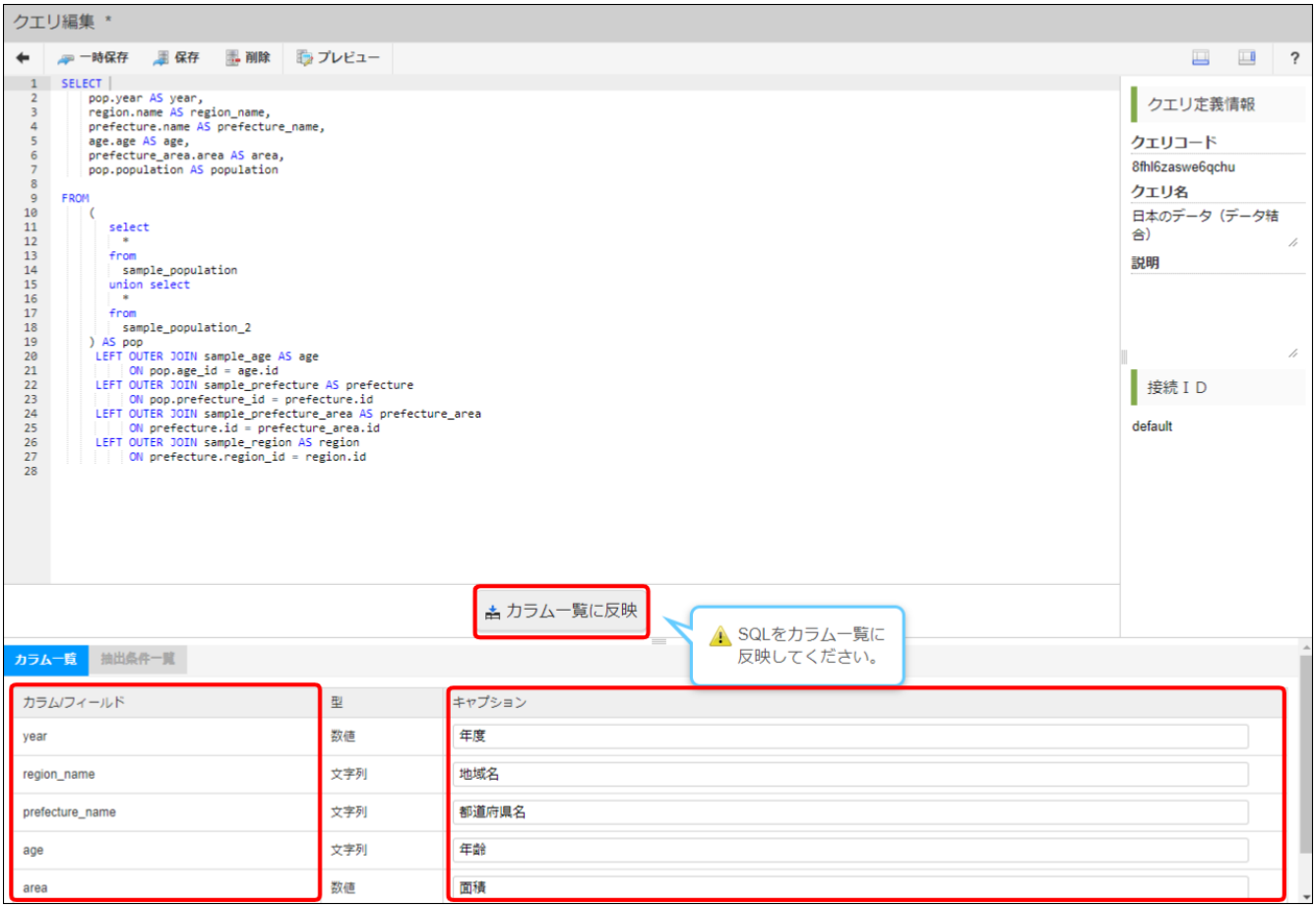
コラム

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。

4. 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。

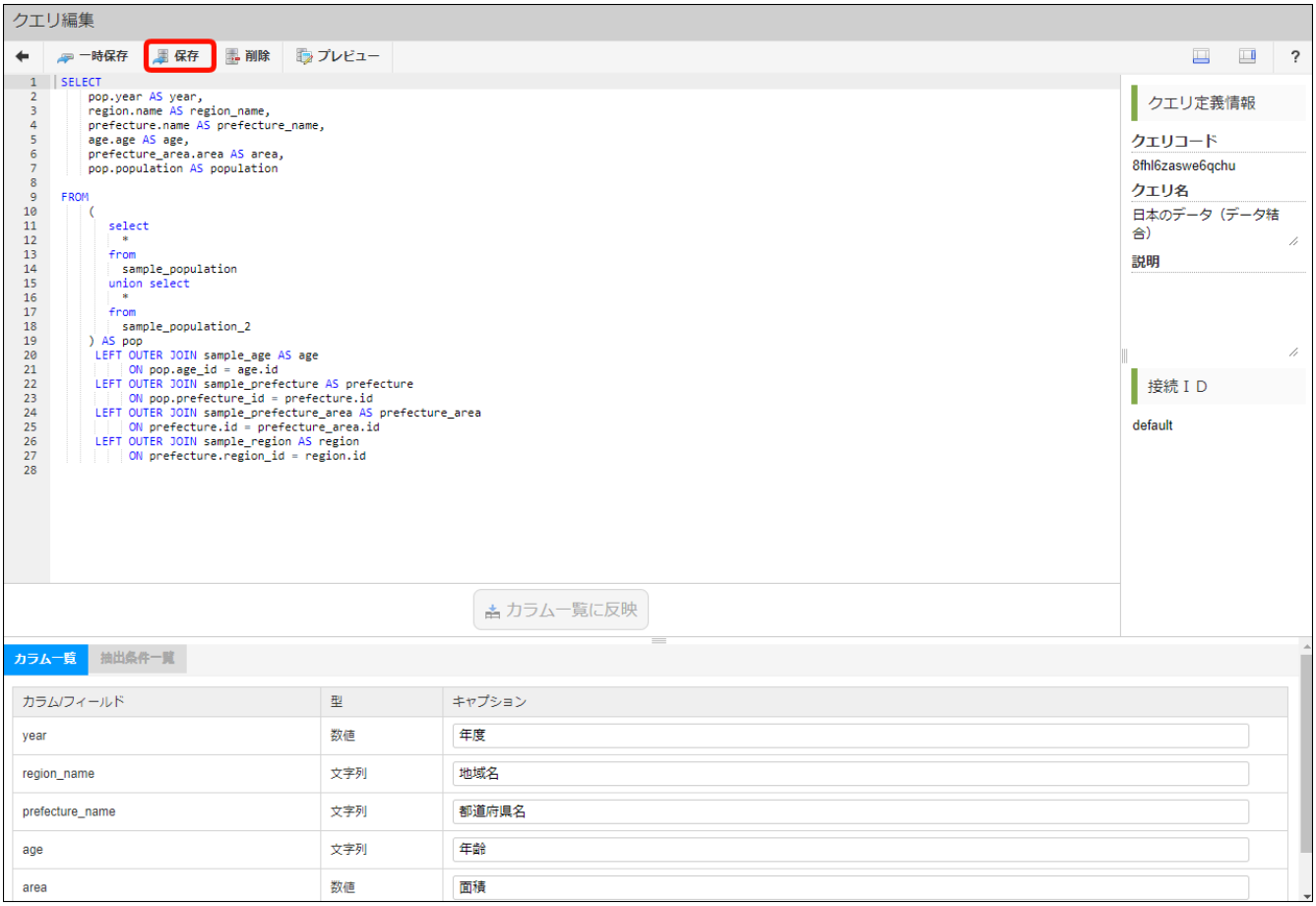
年度	地域名	都道府県名	年齢	面積	人口
2015	九州	宮崎	0 - 14歳	6684.81	150
1990	東北	山形	15 - 64歳	7394.38	819
1970	九州	福岡	15 - 64歳	4843.11	2792
1985	関東	東京	65歳以上	2102.39	1056
2010	中部	福井	15 - 64歳	4189.22	491
1980	東北	山形	65歳以上	7394.38	147
1970	九州	佐賀	0 - 14歳	2439.54	215
2010	四国	愛媛	15 - 64歳	5677.03	865
2005	関東	東京	65歳以上	2102.39	2325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

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カラムは、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。

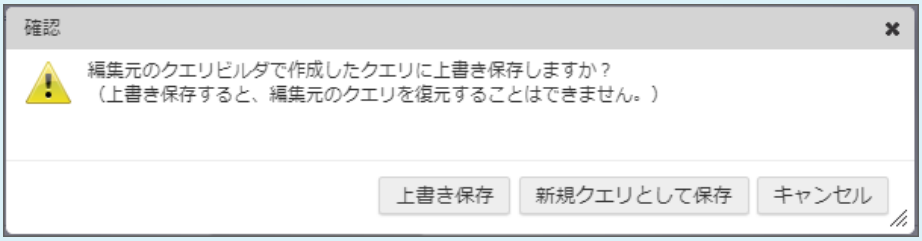
図：クエリ編集画面 - 保存ダイアログ

項目名	説明

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。

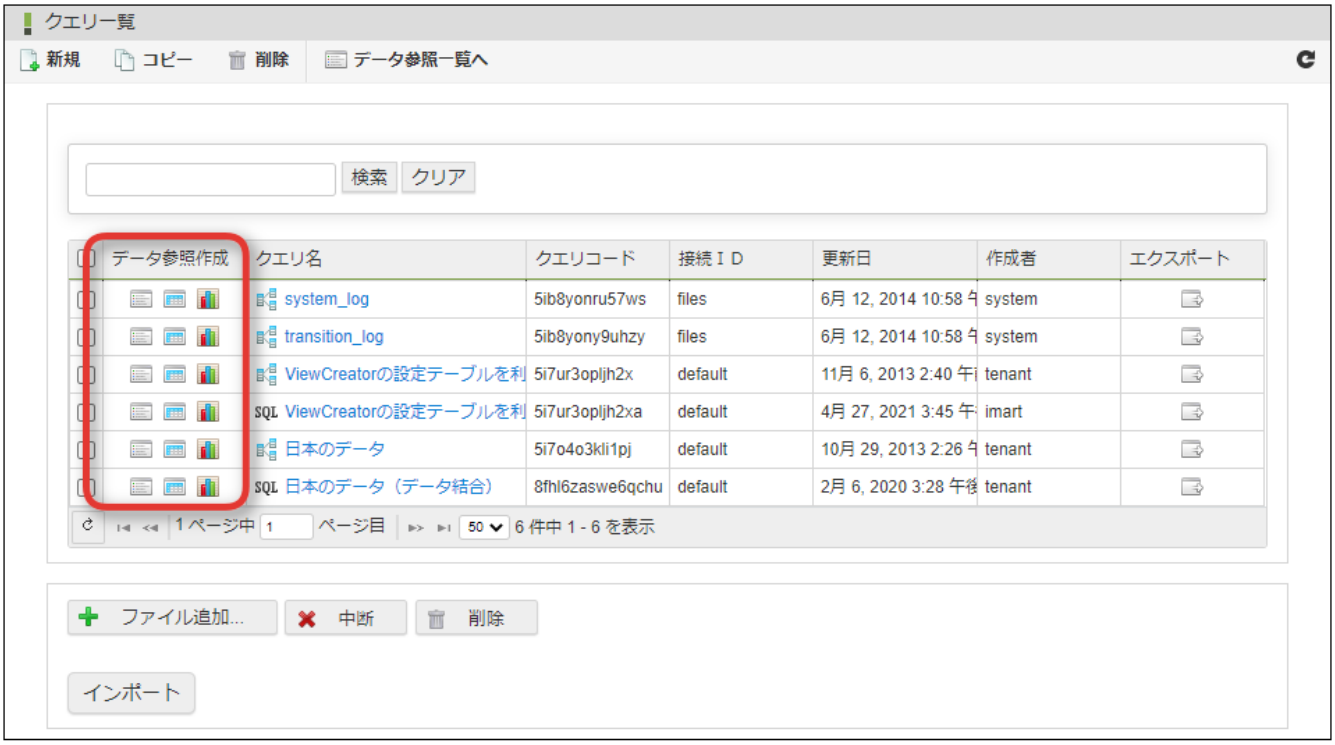
目次

- 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
  - 検索タイプについて
  - 入力制限について
  - AND検索とOR検索（各項目を結合する演算子）について
  - 参照画面の操作状態について
  - 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。



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。
	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。

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。

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

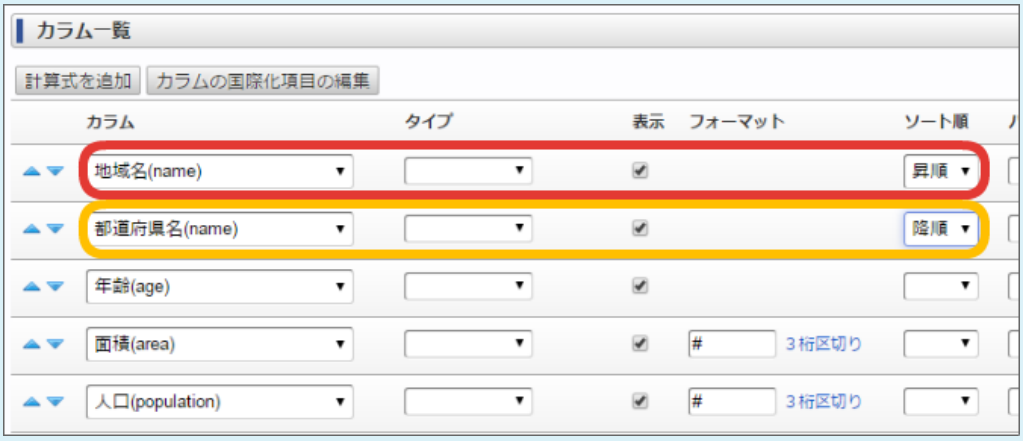
コラム

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。

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。



※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

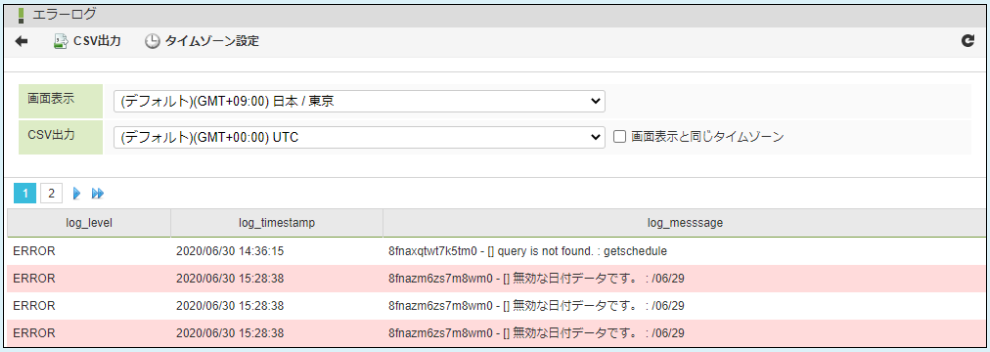
地域名	都道府県名	年齢	面積	人口
関東	栃木	65歳以上	6408	345
関東	栃木	0 - 14歳	6408	307
関東	栃木	15 - 64歳	6408	1352
関東	東京	15 - 64歳	2102	8686
関東	東京	0 - 14歳	2102	1421
関東	東京	65歳以上	2102	1910
関東	千葉	65歳以上	4996	837
関東	千葉	0 - 14歳	4996	843
関東	千葉	15 - 64歳	4996	4236
関東	神奈川	15 - 64歳	2416	6121
関東	神奈川	0 - 14歳	2416	1184
関東	神奈川	65歳以上	2416	1170
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(Zephirine)から利用可能です。

- タイムゾーン設定の表示例

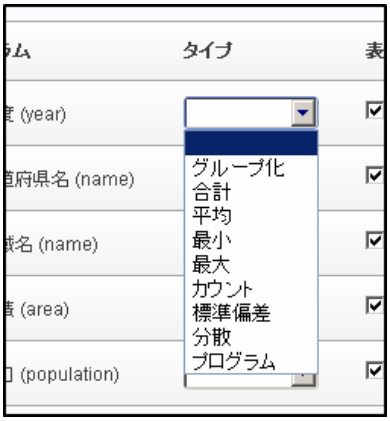


注意

- 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。



- 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

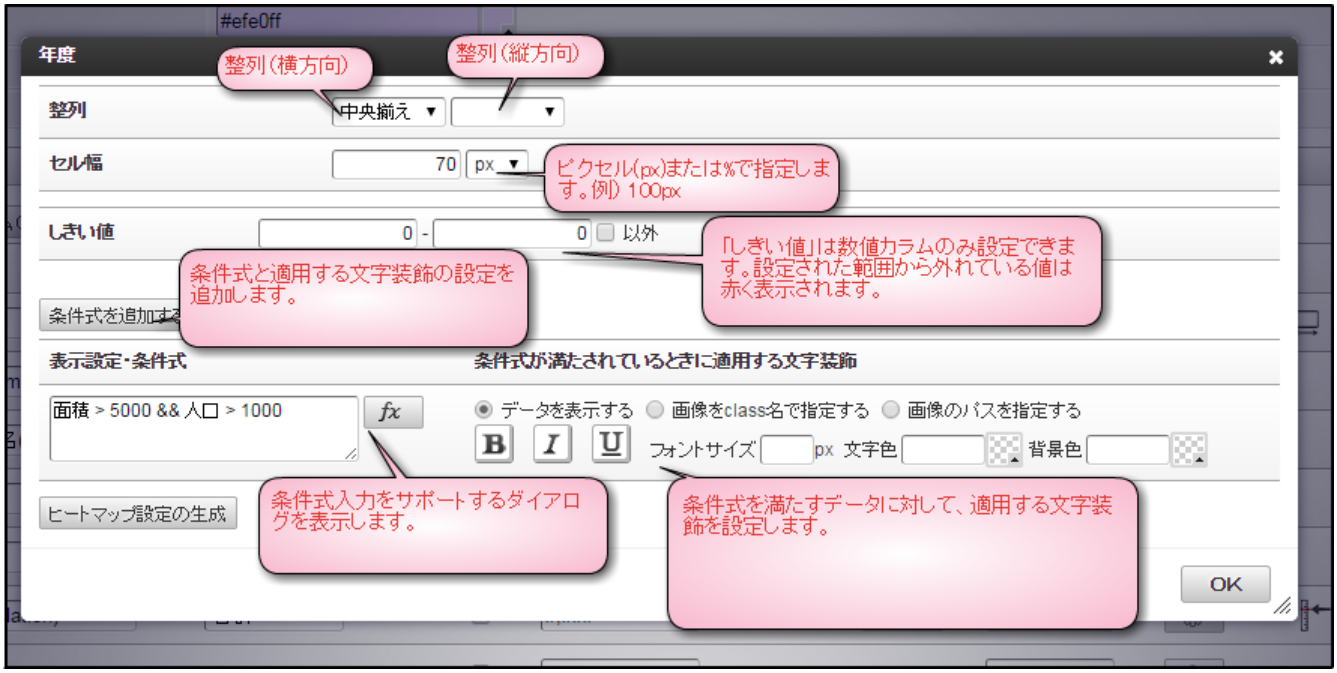
任意のプログラム（ページ）へ連携します。
詳細は応用編「*外部ページへの連携（リスト/サマリ集計のみ）*」を参照してください。

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。

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。

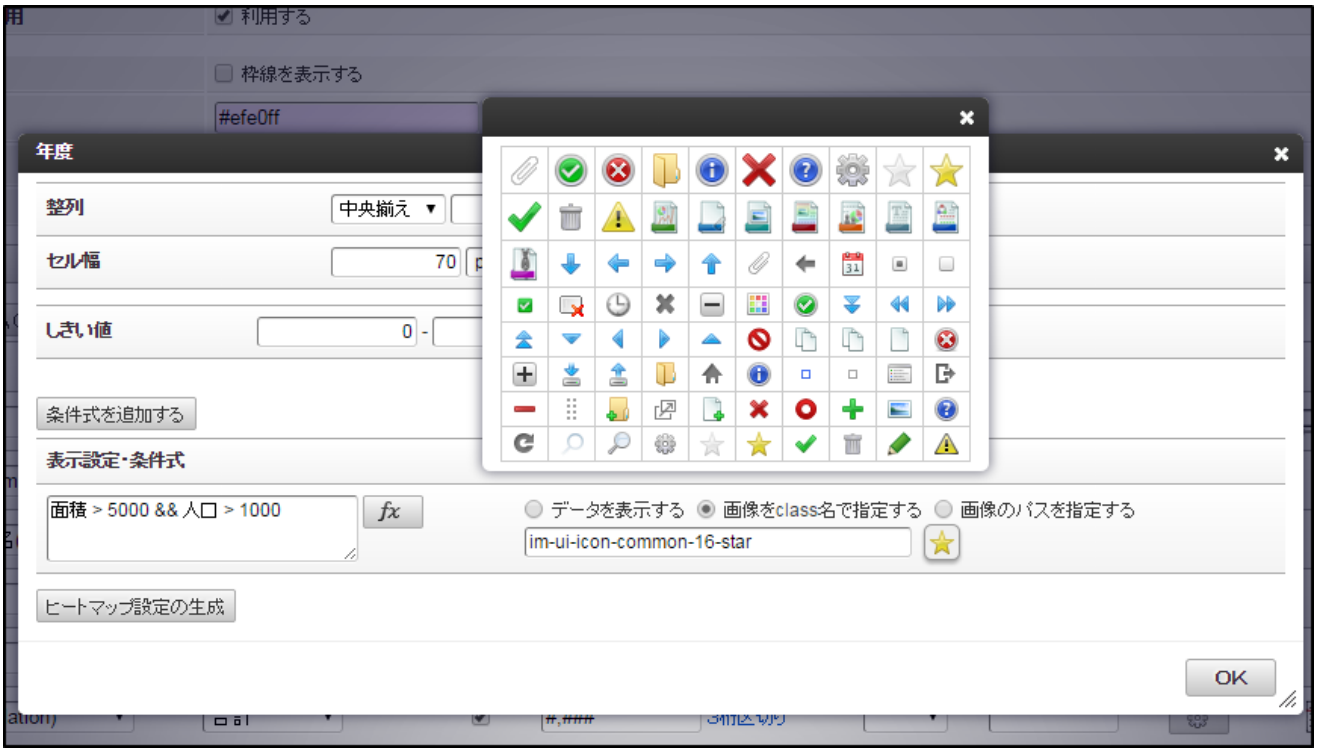
条件式は手入力することも可能ですが、式の作成を補助するダイアログを利用することも可能です。

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。

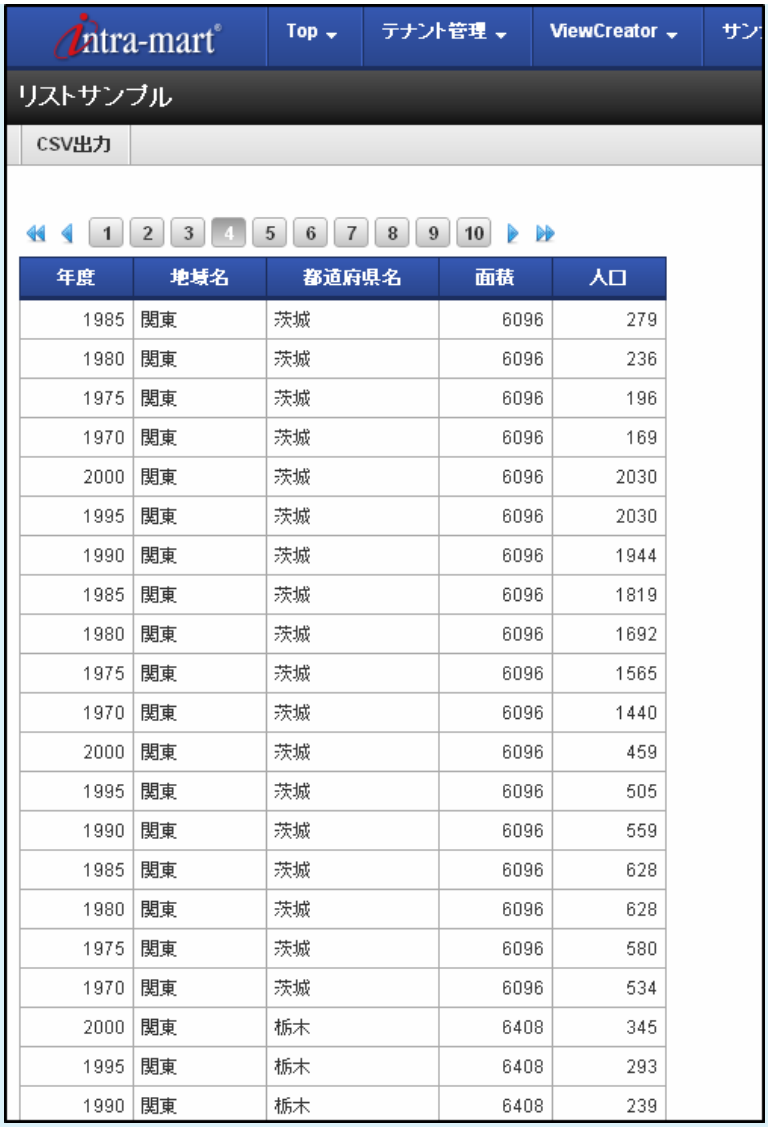


アイコン画像は一覧から選ぶだけでなく、class名や画像のパス名を直接指定できます。

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。



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。

このように表示されます。

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。

実行結果

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。

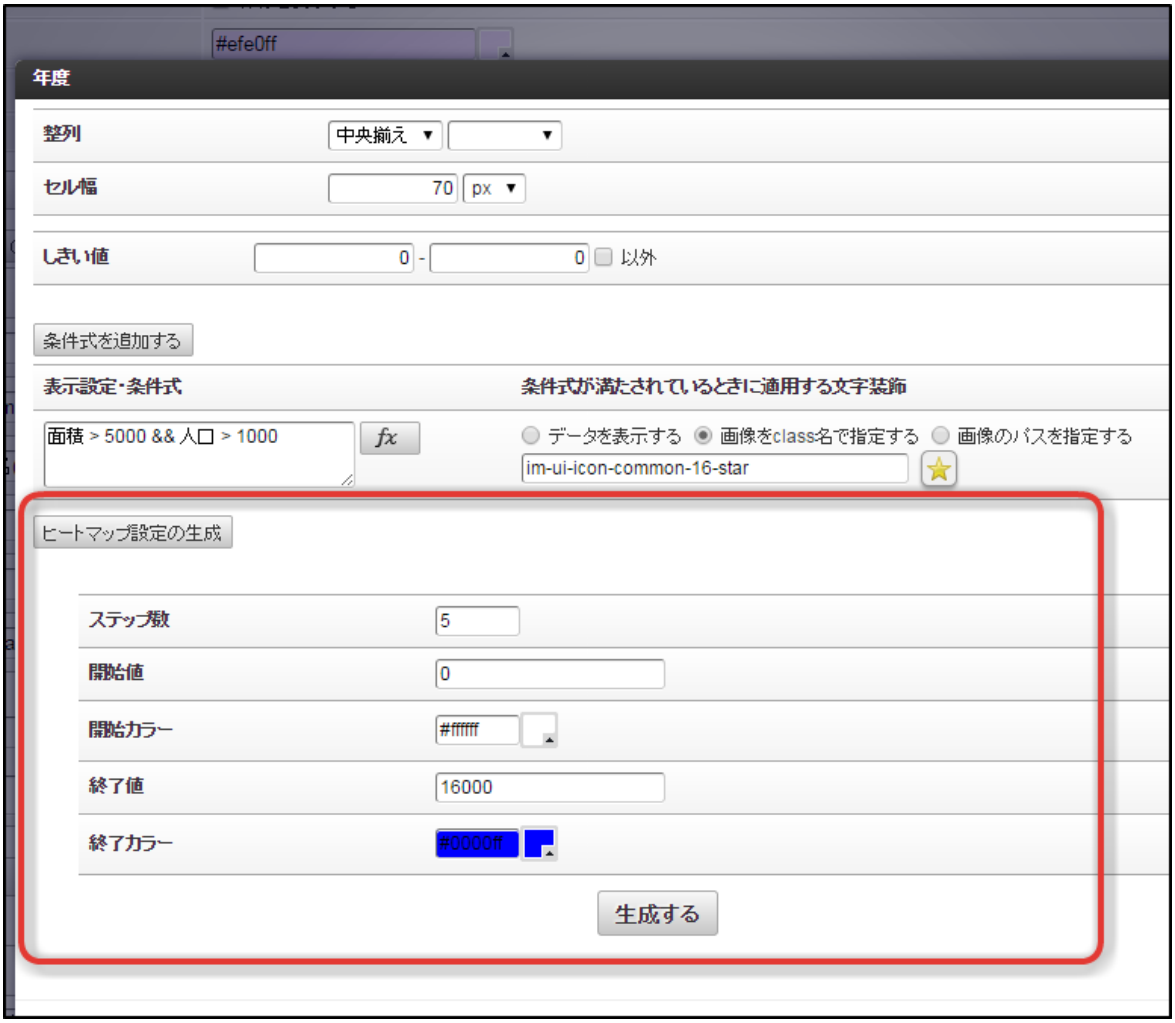
コラム

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

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。



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。

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。

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。

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。

コラム

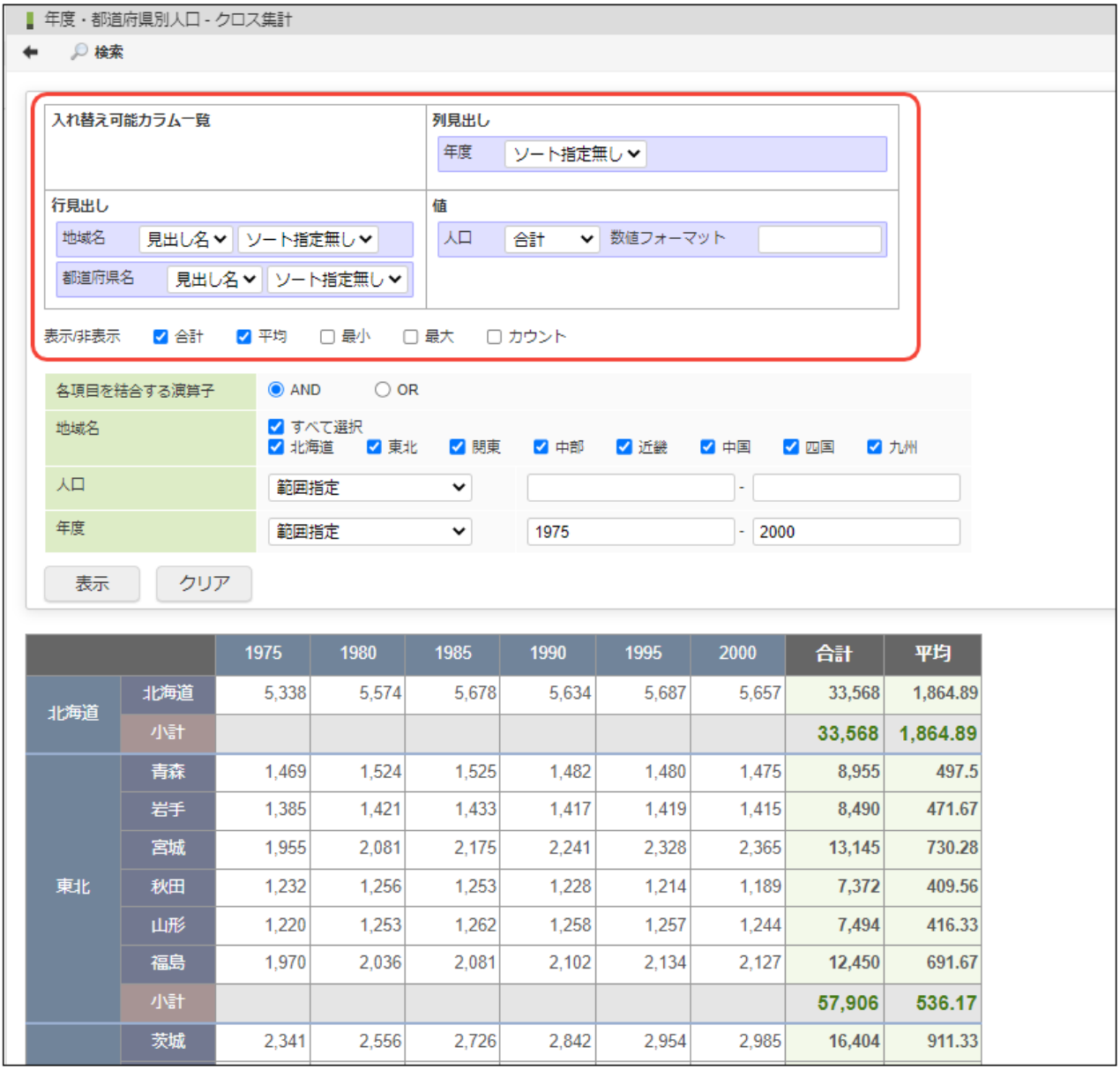
CSVファイルのヘッダには、参照元クエリのカラムで設定されているキャプションが出力されます。

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。

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。



		1975	1980	1985	1990	1995	2000	合計	平均
北海道	北海道	5,338	5,574	5,678	5,634	5,687	5,657	33,568	1,864.89
	小計							33,568	1,864.89
東北	青森	1,469	1,524	1,525	1,482	1,480	1,475	8,955	497.5
	岩手	1,385	1,421	1,433	1,417	1,419	1,415	8,490	471.67
	宮城	1,955	2,081	2,175	2,241	2,328	2,365	13,145	730.28
	秋田	1,232	1,256	1,253	1,228	1,214	1,189	7,372	409.56
	山形	1,220	1,253	1,262	1,258	1,257	1,244	7,494	416.33
	福島	1,970	2,036	2,081	2,102	2,134	2,127	12,450	691.67
	小計							57,906	536.17
	茨城	2,341	2,556	2,726	2,842	2,954	2,985	16,404	911.33

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（カテゴライズされている場合）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（カテゴライズされている場合）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。

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。

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。

コラム

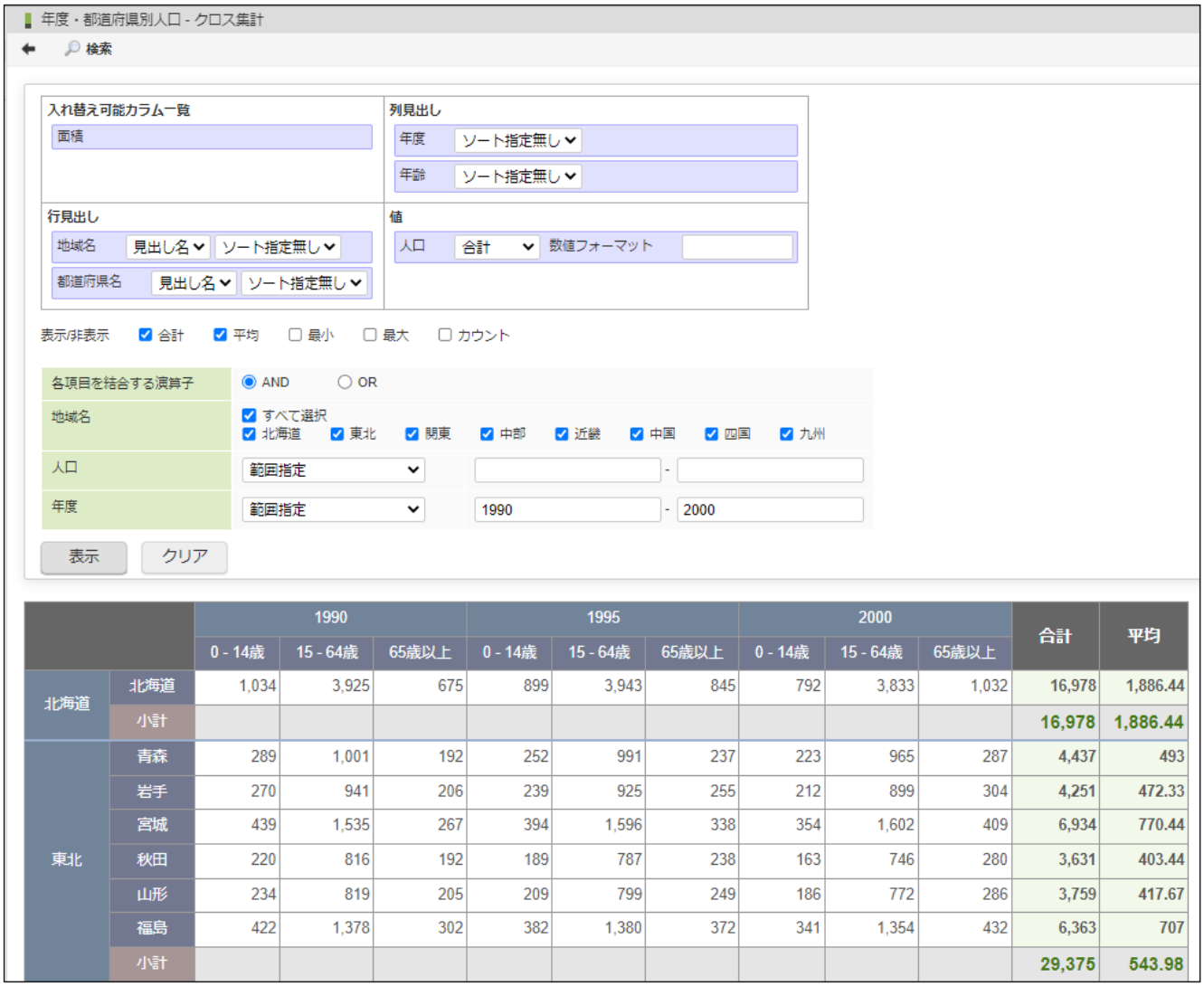
「数値フォーマット」は「値」として設定されたカラムと、「表示/非表示」の下に表示されている入力項目の2箇所で設定できます。
後者の設定は、前者が未設定のときにのみ利用されます。

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: #.##

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。



		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。

コラム
メモリの使用量については クロス集計設定 の設定を見直すことで抑制できます。

グラフ集計の作成

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

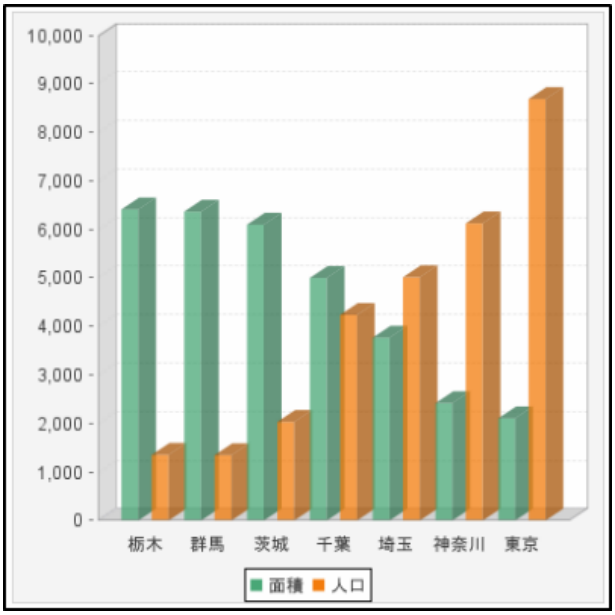


図 棒グラフイメージ

折れ線グラフ

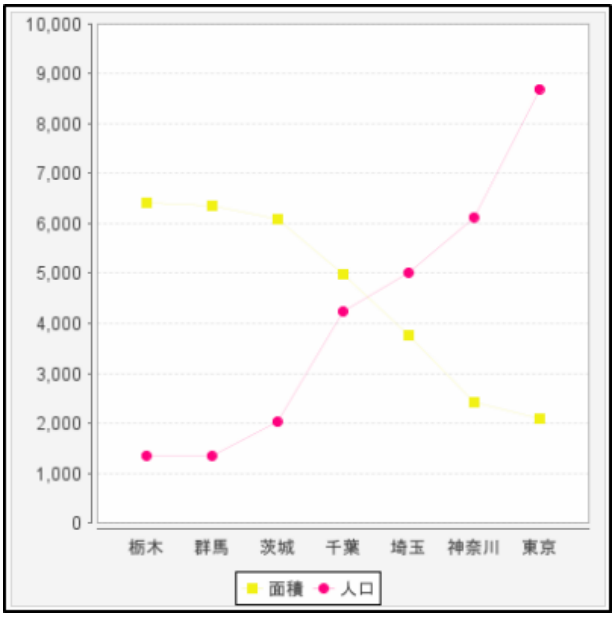


図 折れ線グラフイメージ

レーダーチャート

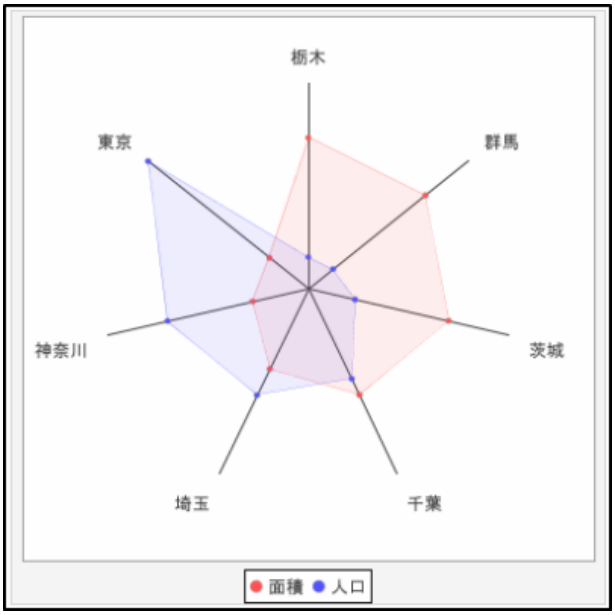


図 レーダーチャートイメージ

円グラフ

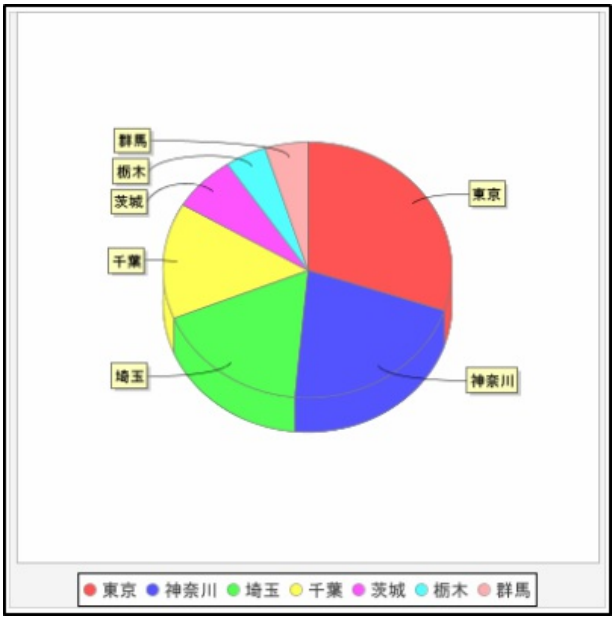


図 円グラフイメージ

グラフ設定

グラフ表示時のレイアウトを設定します。

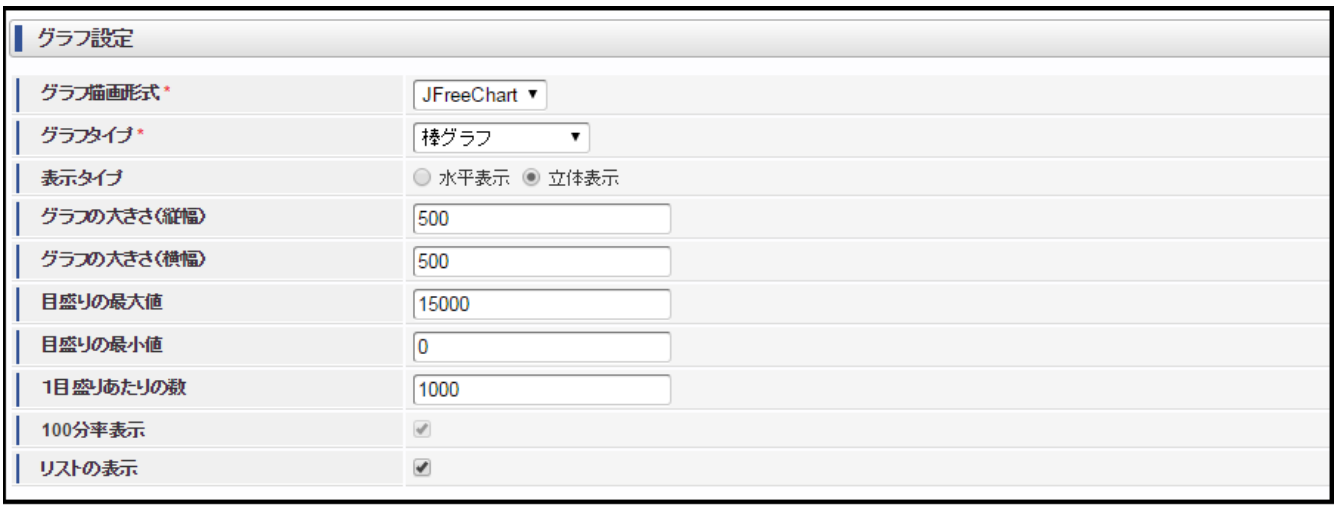


図 データ参照・編集画面 グラフ設定項目

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。

項目名	説明
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。

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。

注意

グラフ集計を作成するためには、最低1つ以上の数値カラムが必要です。

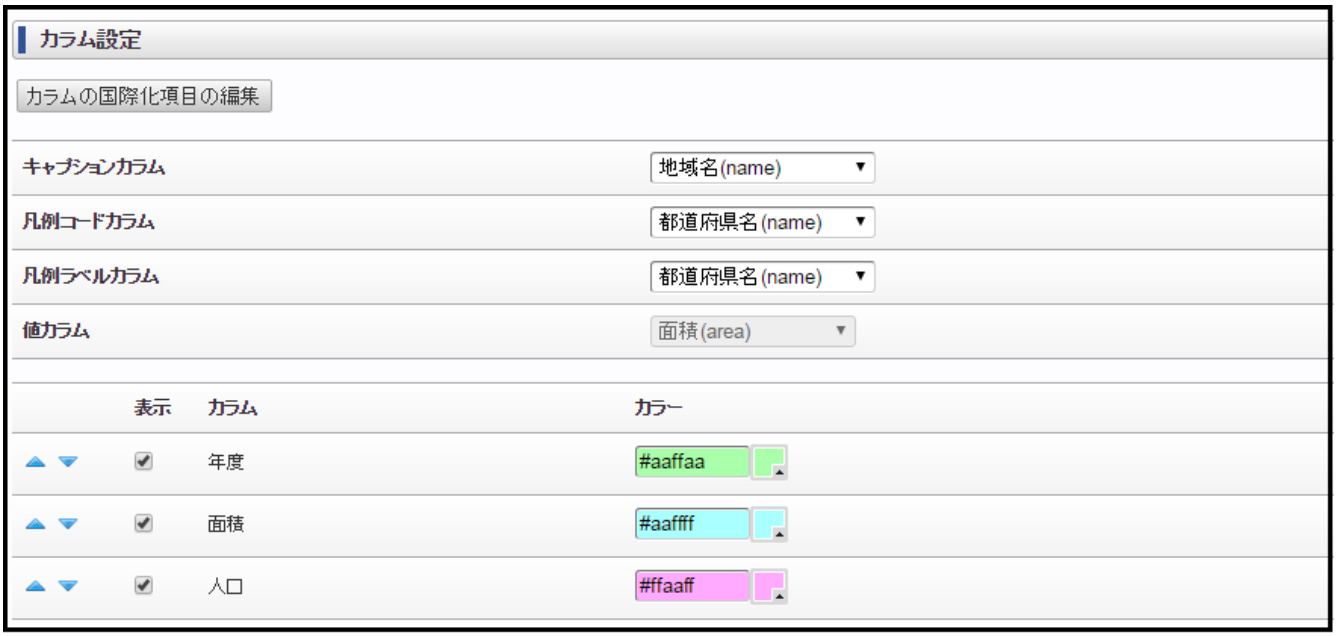


図 データ参照・編集画面 カラム設定項目

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、「*国際化とヘッダ・フッタ*」 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、「*グラフ集計で「凡例」の表示方法を変更する*」 を参照してください。
凡例ラベルカラム	詳細は、「*グラフ集計で「凡例」の表示方法を変更する*」 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。

項目名	説明
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。

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て作成できます。

コラム

URL（2019年6月現在）
Highcharts ： https://www.highcharts.com/

グラフ

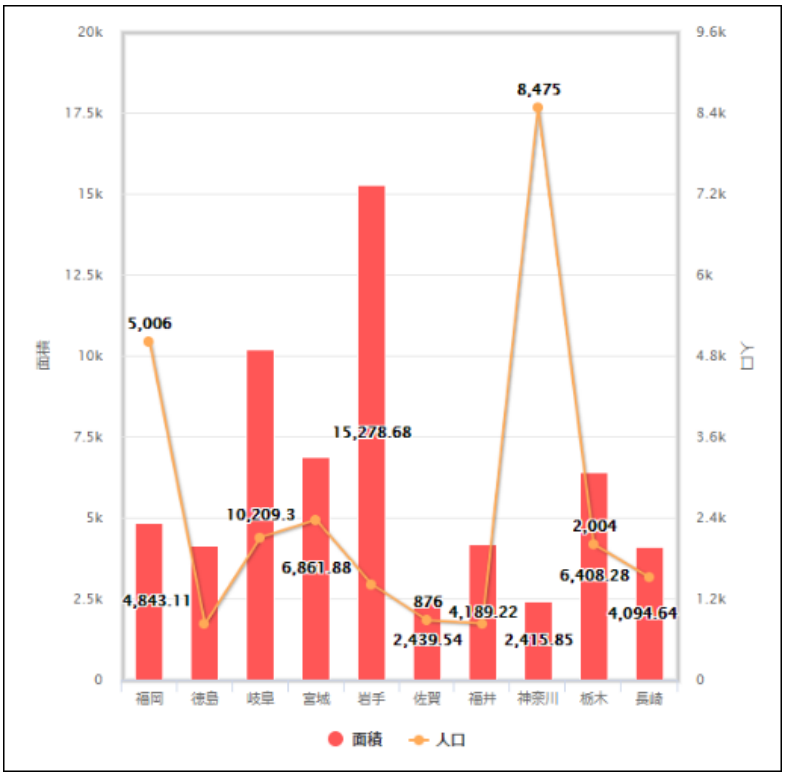


図 グラフ組み合わせイメージ

棒グラフ

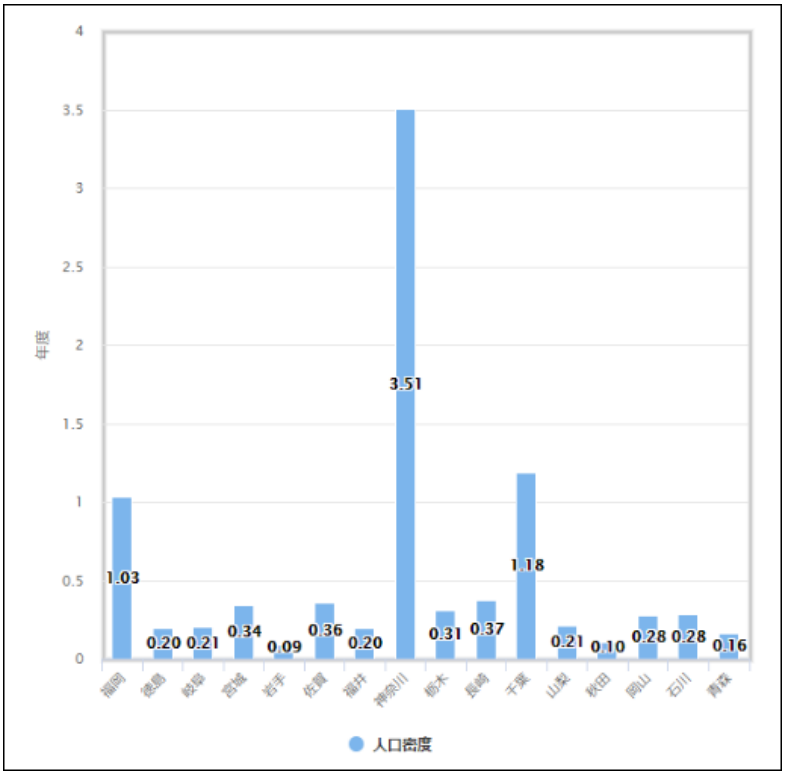


図 棒グラフイメージ

折れ線グラフ

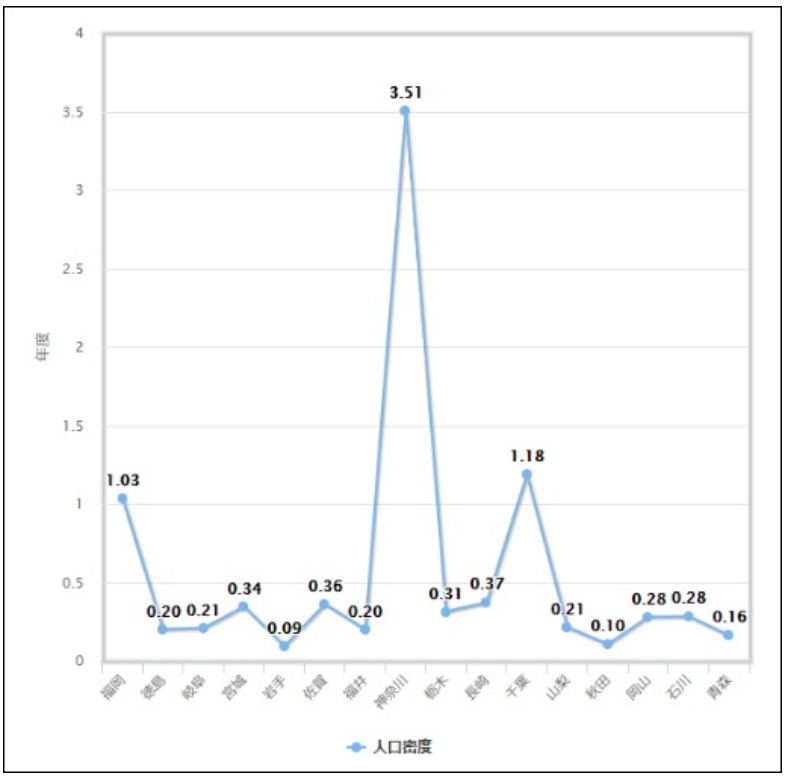


図 折れ線グラフイメージ

レーダーチャート

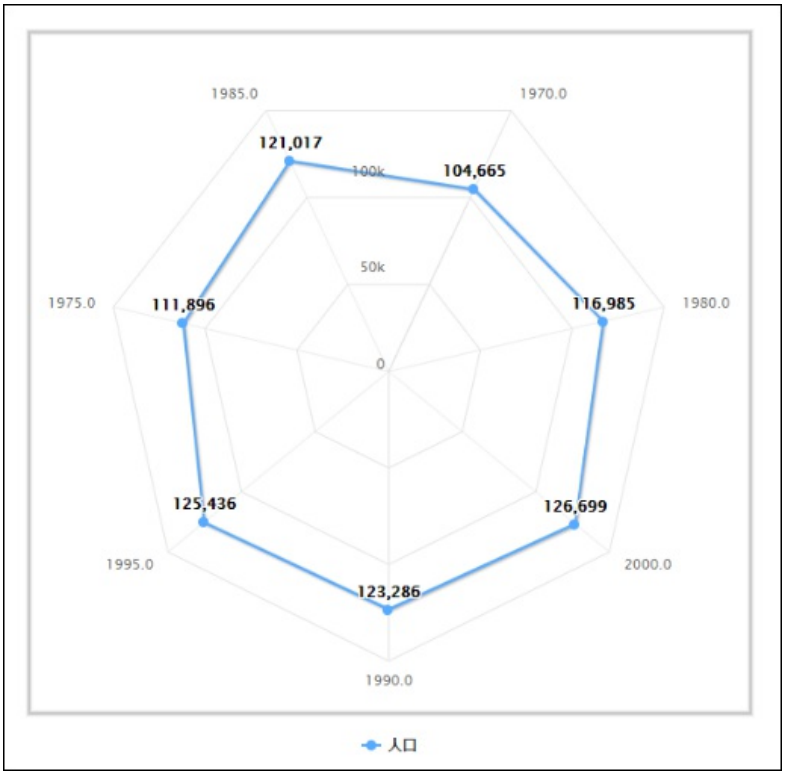


図 レーダーチャートイメージ

円グラフ

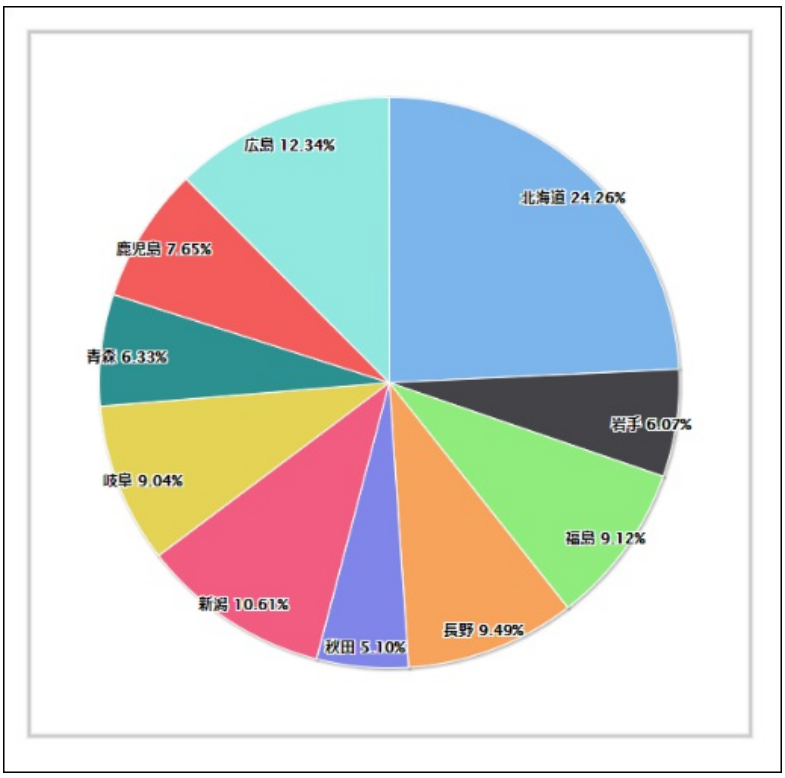


図 円グラフイメージ

エリアチャート

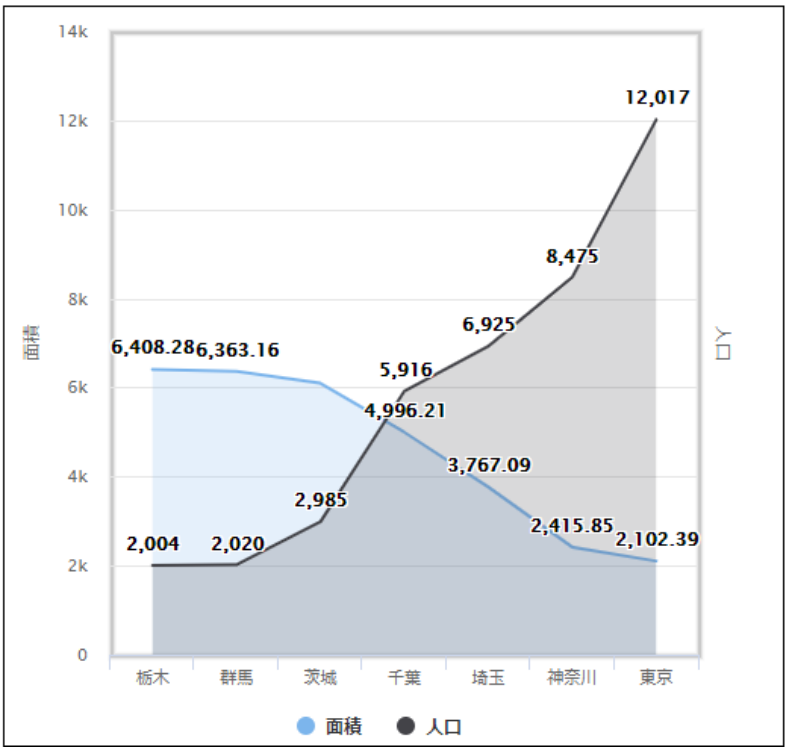


図 エリアチャートイメージ

グラフ設定

グラフ表示時のレイアウトを設定します。

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。

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、「*国際化とヘッダ・フッタ*」 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「*レーダーチャート*」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、「*グラフ集計で「凡例」の表示方法を変更する*」 を参照してください。
凡例ラベルカラム	詳細は、「*グラフ集計で「凡例」の表示方法を変更する*」 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「*棒グラフ*」、「*折れ線グラフ*」、「*レーダーチャート*」、「*円グラフ*」 、「*エリアチャート*」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に設定できます。

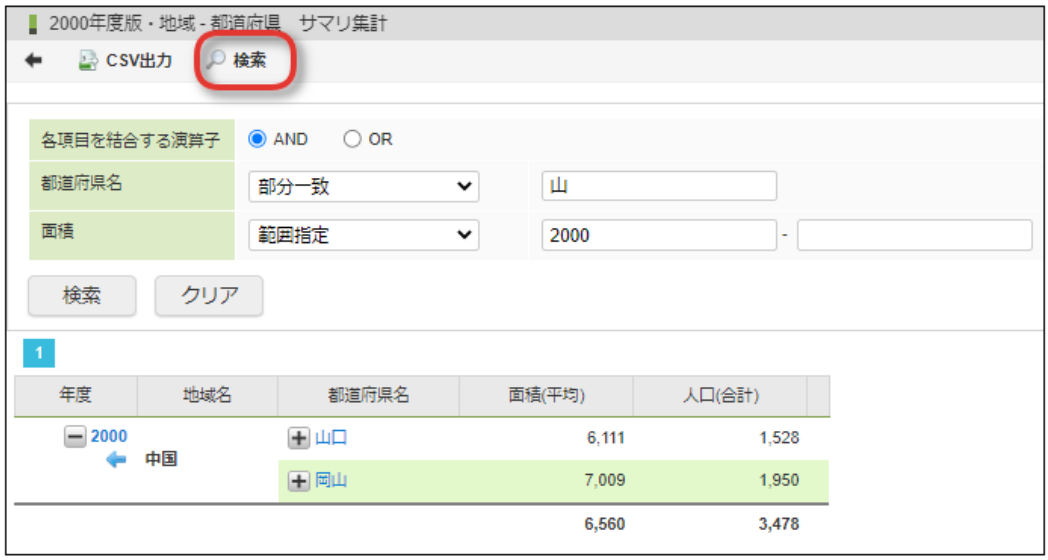
コラム

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

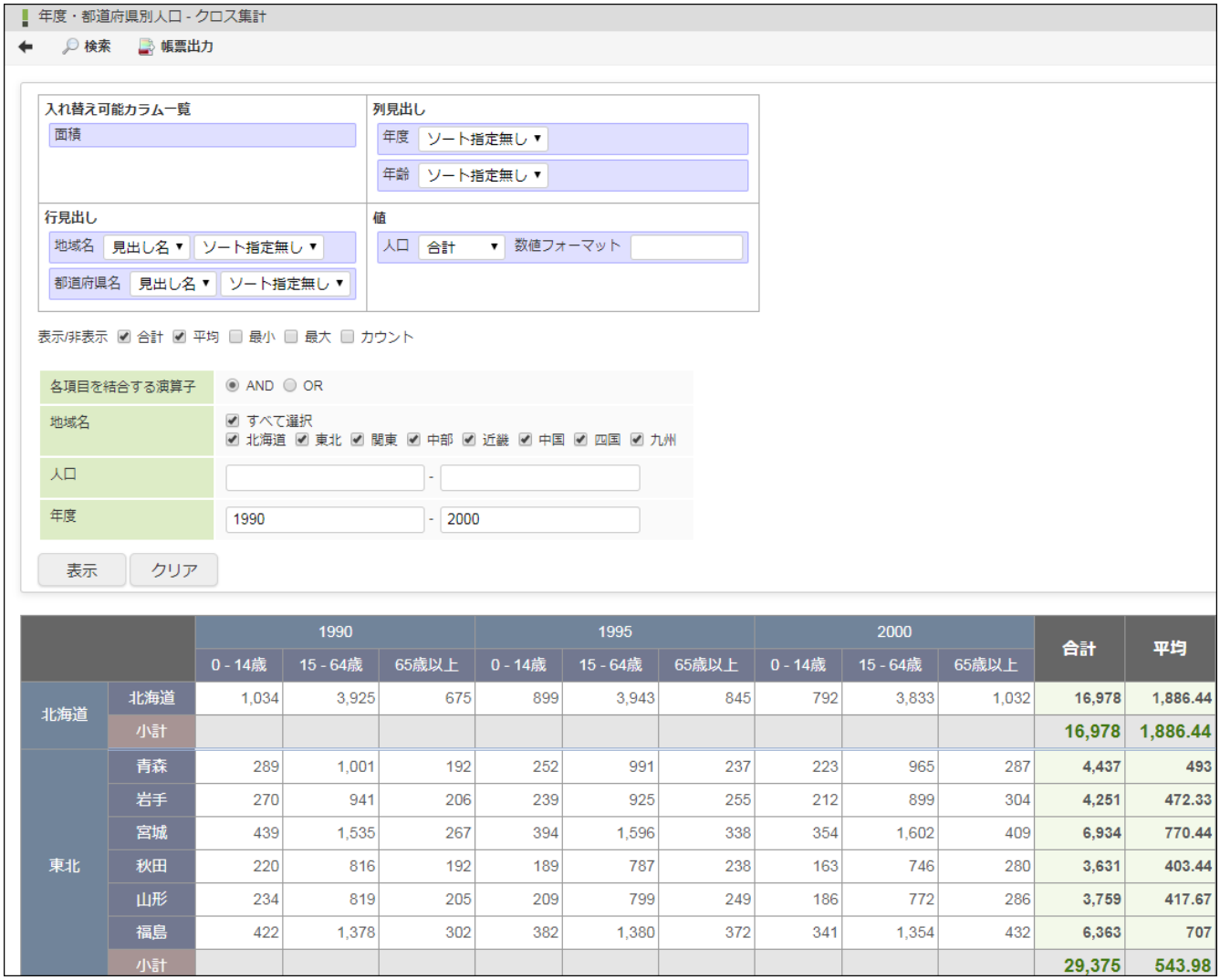
1.追加ボタン	検索設定を 1 つ追加します。
2.各項目を結合する演算子	AND検索または、OR検索を設定します。 詳細は「*AND検索とOR検索（各項目を結合する演算子）について*」を参照してください。
3.参照画面の操作状態	データ参照表示画面で入力された検索条件値の維持に関する設定です。 詳細は「*参照画面の操作状態について*」を参照してください。
4.削除ボタン	検索設定を削除します。
5.上へ、下へボタン	検索設定の順序を変更します。 ドラッグ&ドロップで順番を入れ替えることも可能です。
6.検索対象カラム	条件値と比較を行う対象とするカラムを設定します。
7.検索タイプ	検索条件値の入力パターンを設定します。 詳細は「*検索タイプについて*」を参照してください。
8.入力制限	データ参照閲覧時における検索条件の入力、変更に関する制限を設定します。 詳細は「*入力制限について*」を参照してください。
9.検索パターン	検索条件値の適用方法を設定します。 詳細は「*検索の条件値入力に選択型コンポーネントを利用する*」を参照してください。
10.検索条件デフォルト値	検索条件値のデフォルト値を設定します。 適用方法については「*検索タイプについて*」を参照してください。
11.検索条件値・選択肢リスト	検索条件値の入力方法としてテキストボックスの代わりに選択型コンポーネントを利用するための設定です。 詳細は「*検索の条件値入力に選択型コンポーネントを利用する*」を参照してください。

検索設定を追加すると、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



- クロス集計の場合



		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

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。

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。

検索タイプについて

検索設定で利用可能な検索タイプ（検索方法）は以下の通りです。

検索タイプ	検索パターン	説明
キーワード検索	部分一致	条件値を含むデータを検索します。 条件値が空の場合は、検索条件から除外されます。
	前方一致	条件値で始まるデータを検索します。 条件値が空の場合は、検索条件から除外されます。
	完全一致	条件値と完全に同一のデータを検索します。 条件値が空の場合は、検索条件から除外されます。
	空に一致	NULLまたは空文字を条件値とします。 「空に一致」をデフォルト値とする場合、「検索条件値・選択肢リスト」は設定できません。
範囲検索	範囲指定	上限値と下限値による検索を行います。 条件値が空の場合は、検索条件から除外されます。 例えば、上限値が空の場合は下限値だけが適用されます。
	NULLに一致	NULLを条件値とします。 外部データソースを利用している場合は選択できません。
リスト選択		チェックボックス形式で条件値の選択を行います。 クロス集計の場合のみ利用可能です。

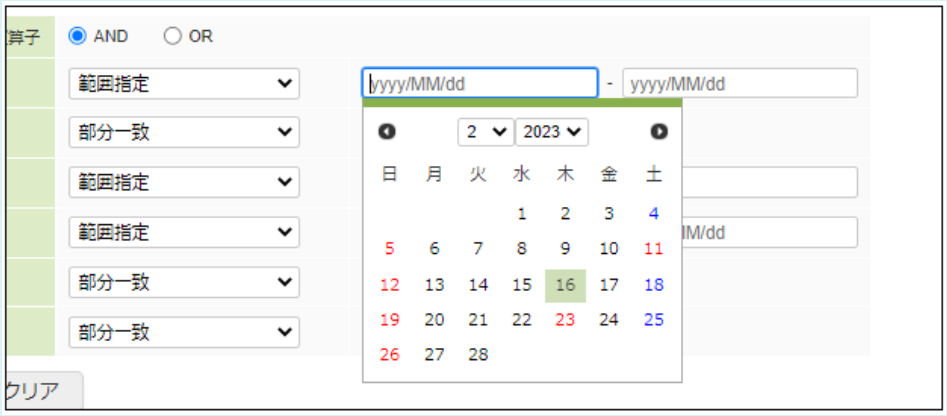
コラム

データ参照を表示したときの動作について

選択型コンポーネントが利用されている検索設定の場合、「空に一致」は選択できません。
また、「セレクトボックス」が設定されている場合は、「検索タイプ」の選択項目自体が表示されません。
選択型コンポーネントについては、「*検索の条件値入力に選択型コンポーネントを利用する*」を参照してください。

コラム

「日付型」と「タイムスタンプ型」のカラムを検索設定で利用する場合、カレンダー入力が可能です。

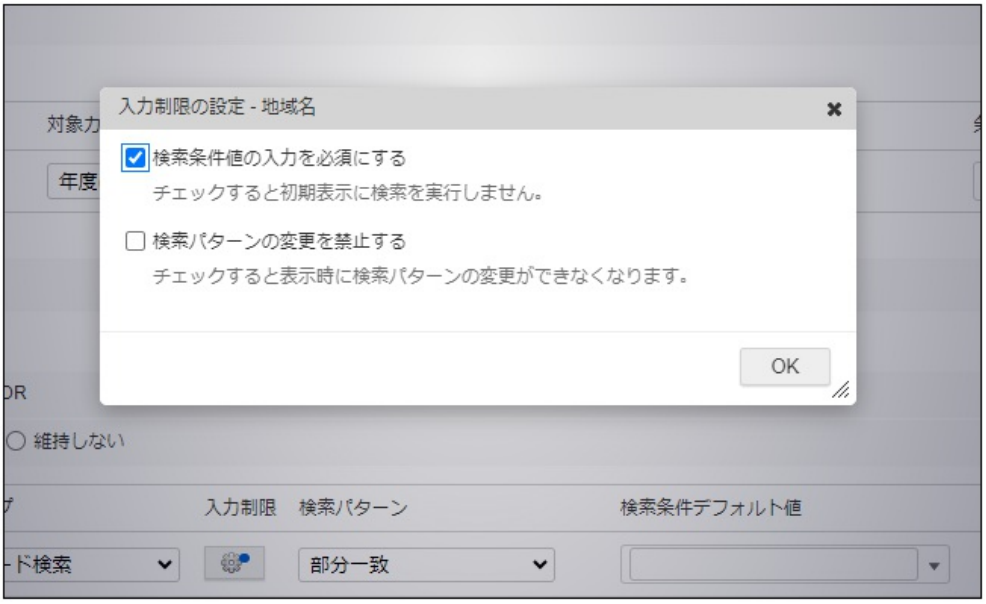


コラム

検索設定のカレンダー入力、「空に一致」と「NULLに一致」は2023 Spring(Gerbera)から利用可能です。

入力制限について

データ参照閲覧時において、「キーワード検索」の検索設定について以下の制限を付加できます。



- 検索条件値の入力を必須にする

検索条件値の入力が無い場合、検索を実行できません。
必須入力の検索設定が１つ以上存在する場合、初期表示時に検索の実行を行いません。
また、必須入力に設定された検索設定は「空に一致」を利用できません。

- 検索パターンの変更を禁止する

閲覧画面において検索パターンの変更ができません。

> コラム
>
> 入力制限は2023 Autumn(Hollyhock)から利用可能です。

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

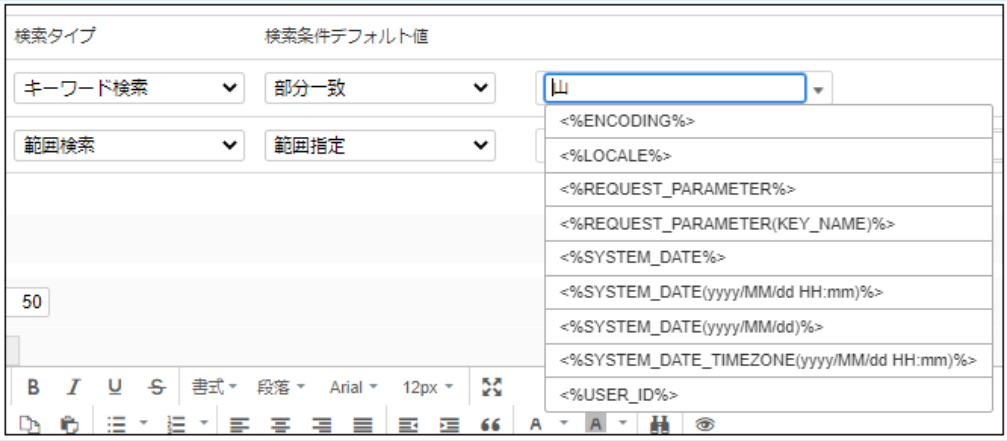
検索項目のうち、いずれかを満たすレコードが検索結果として表示されます。
ただし、条件値が入力されていない検索項目を除きます。

> 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。

> 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。

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。



コラム

OR検索は2019 Spring(Violette)より、利用できます。

参照画面の操作状態について

- 維持する場合

最後に表示したデータの検索条件やソート順、グループ化された列の展開状況等の操作状態が保存されます。
保存された操作状態は「ブラウザを閉じる」、「リロードアイコンをクリックする」、「データ参照定義の変更が行われる」のいずれかの操作によって破棄されます。

- 維持しない場合

検索条件やソート順、グループ化された列の展開状況等の操作状態について保存や明示的な破棄を行いません。
そのため、リロードアイコンをクリックした場合、画面の操作状態に変化はありません。
他の画面に遷移したタイミングで操作状態に関する情報は失われます。

コラム

参照画面の操作状態は2023 Spring(Gerbera)より、設定できます。

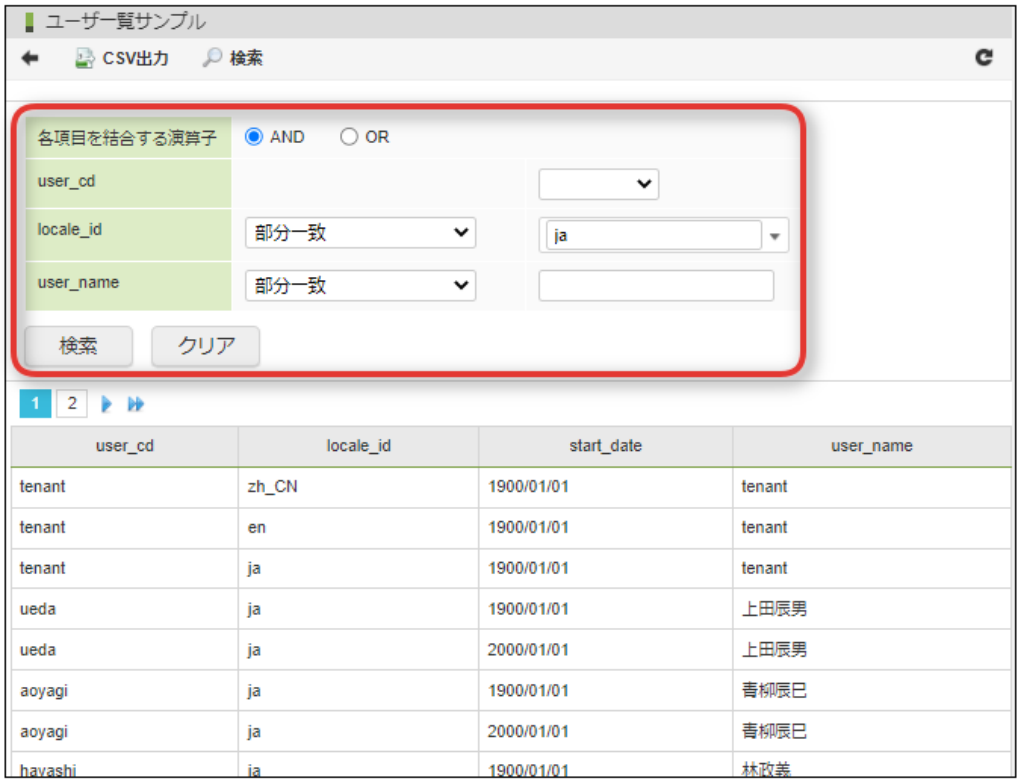
リスト集計とサマリ集計で利用可能な機能

検索の条件値入力に選択型コンポーネントを利用する

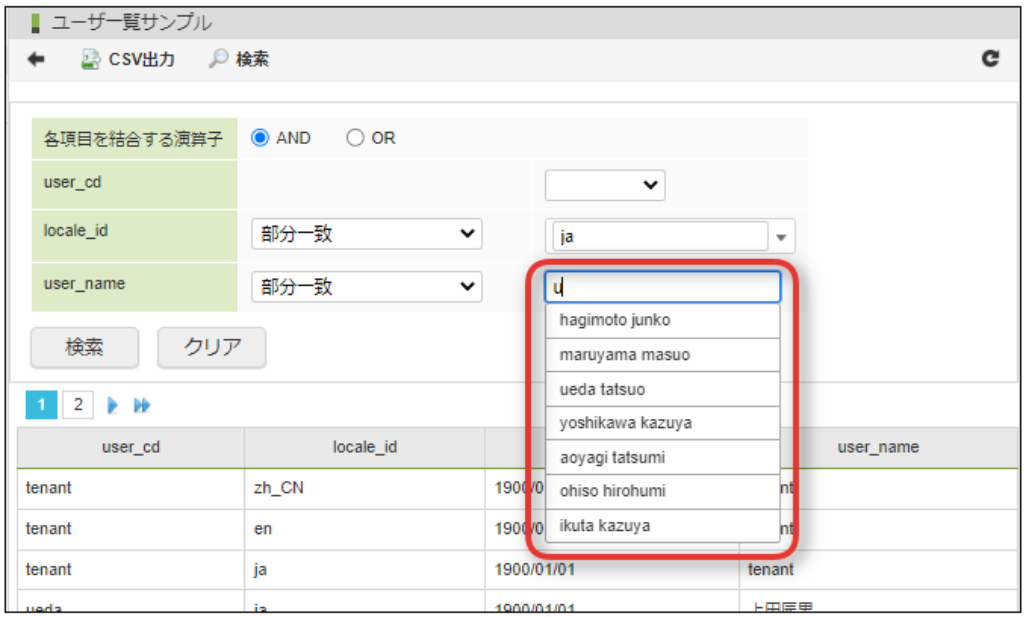
検索条件値はテキストフィールドでの入力だけでなく、「セレクトボックス」、「コンボボックス」、「オートコンプリート」を利用できます。

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。



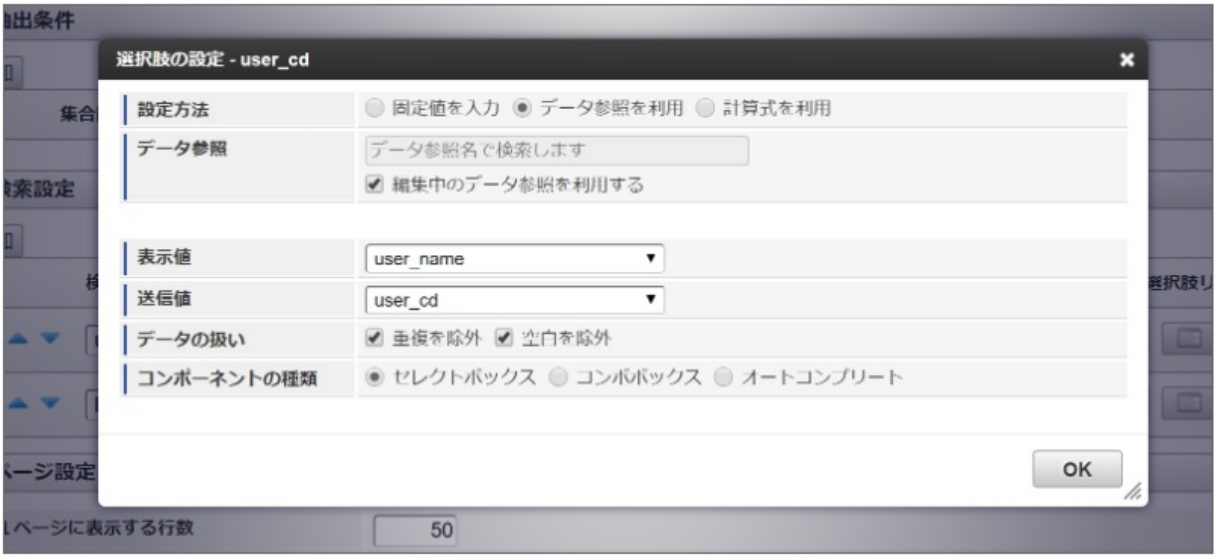
オートコンプリートを利用した場合の例。



- 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。

「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。



- 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。

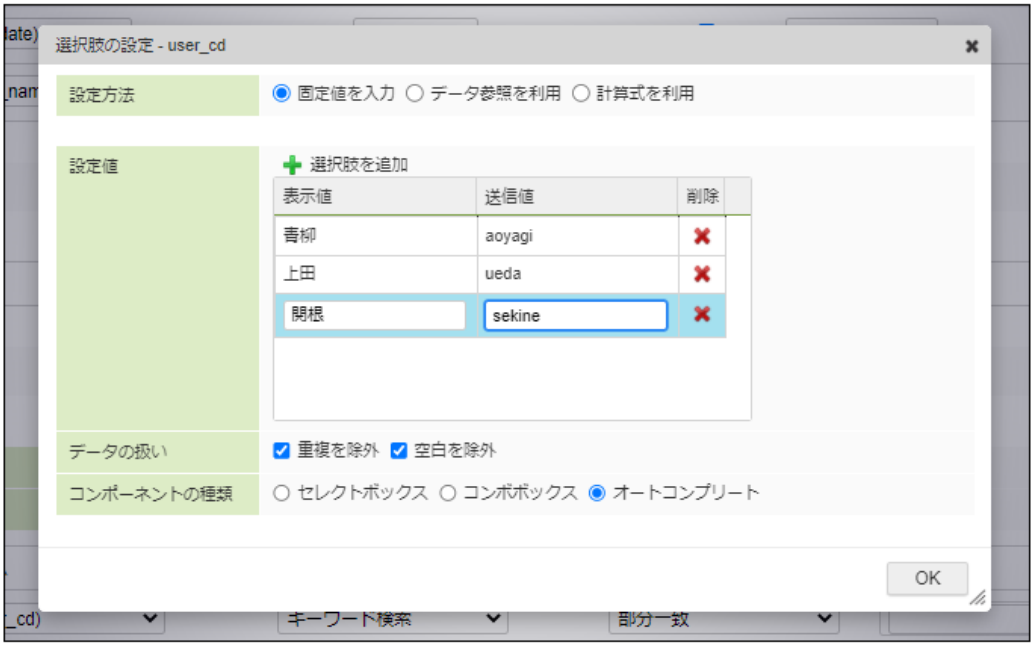
- 「データ参照を利用」が選択された場合の設定

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。

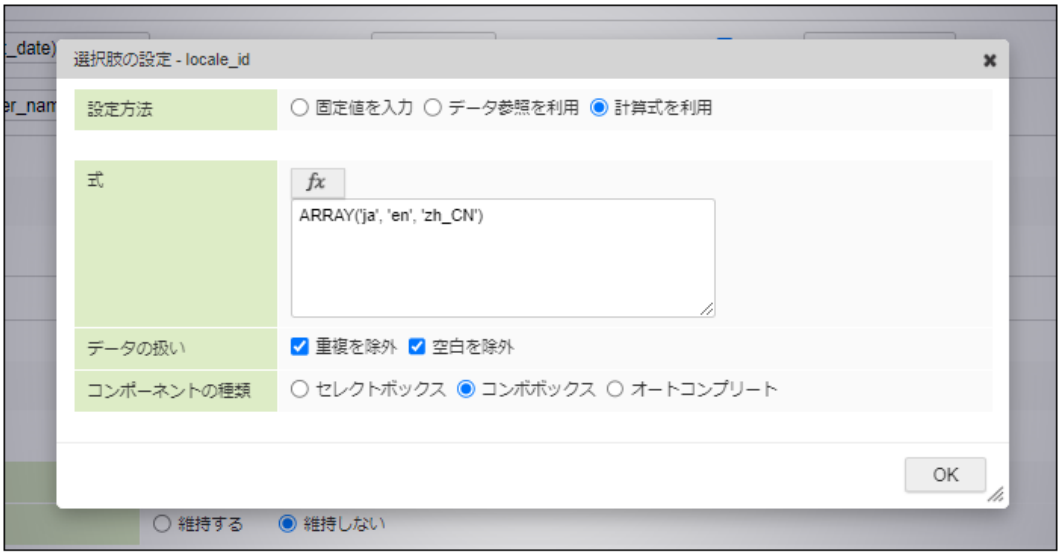
- 「固定値を入力」が選択された場合の設定

「表示値」と「送信値」を手入力で設定します。

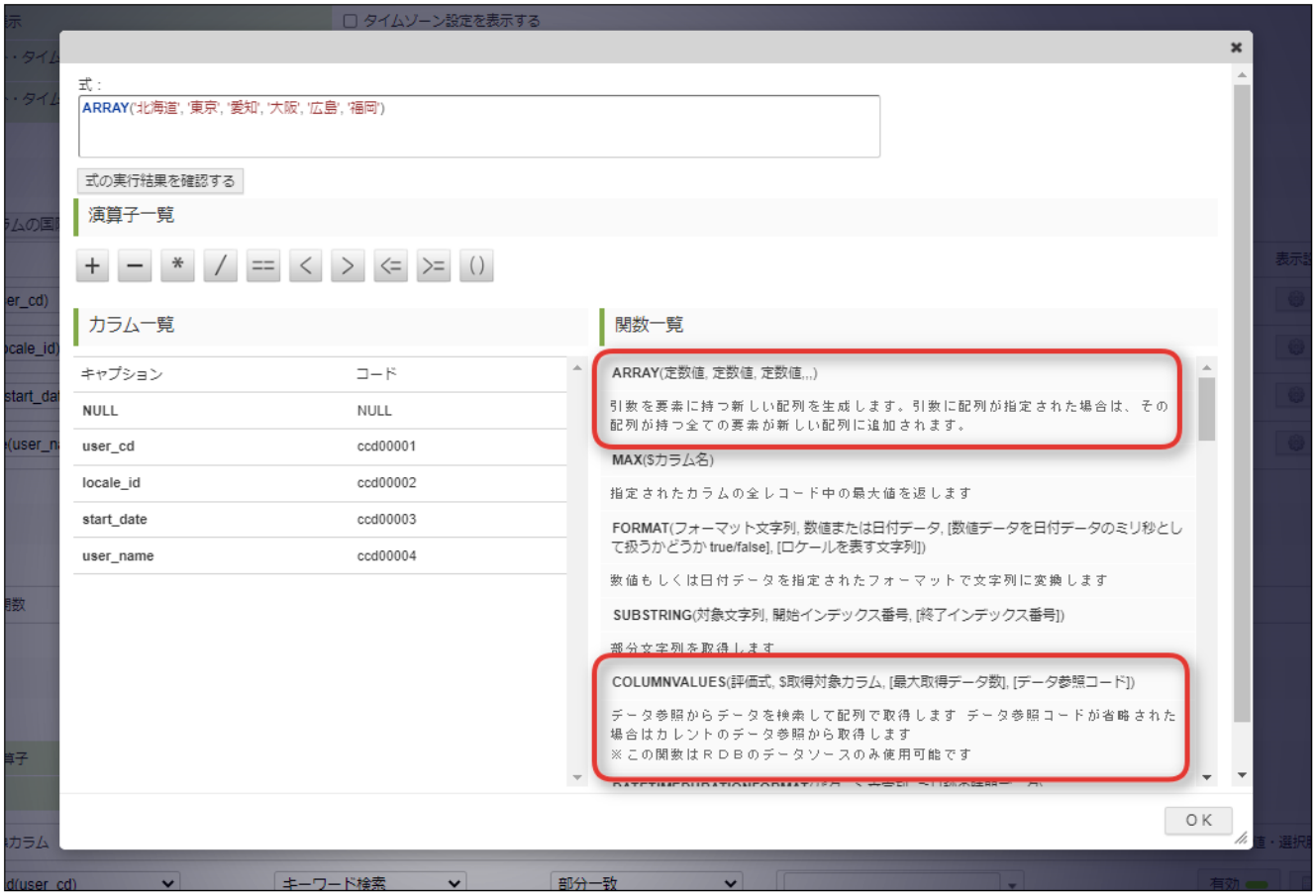


- 「計算式を利用」が選択された場合の設定

「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。

- 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。

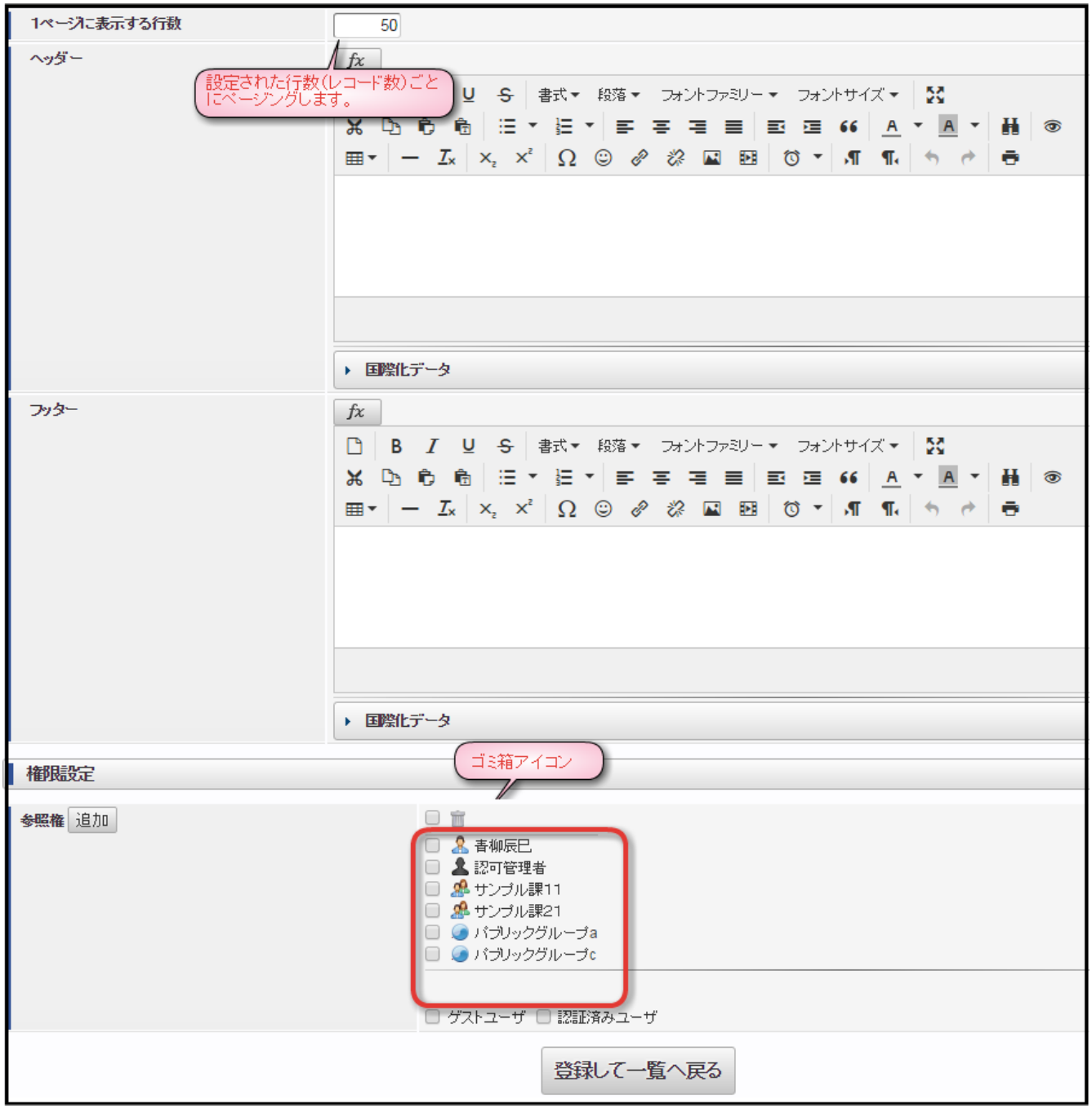
> **注意**
>
> 日付型、タイムスタンプ型、バイナリ型、真偽値型のカラムは「リスト選択」では利用できません。

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。

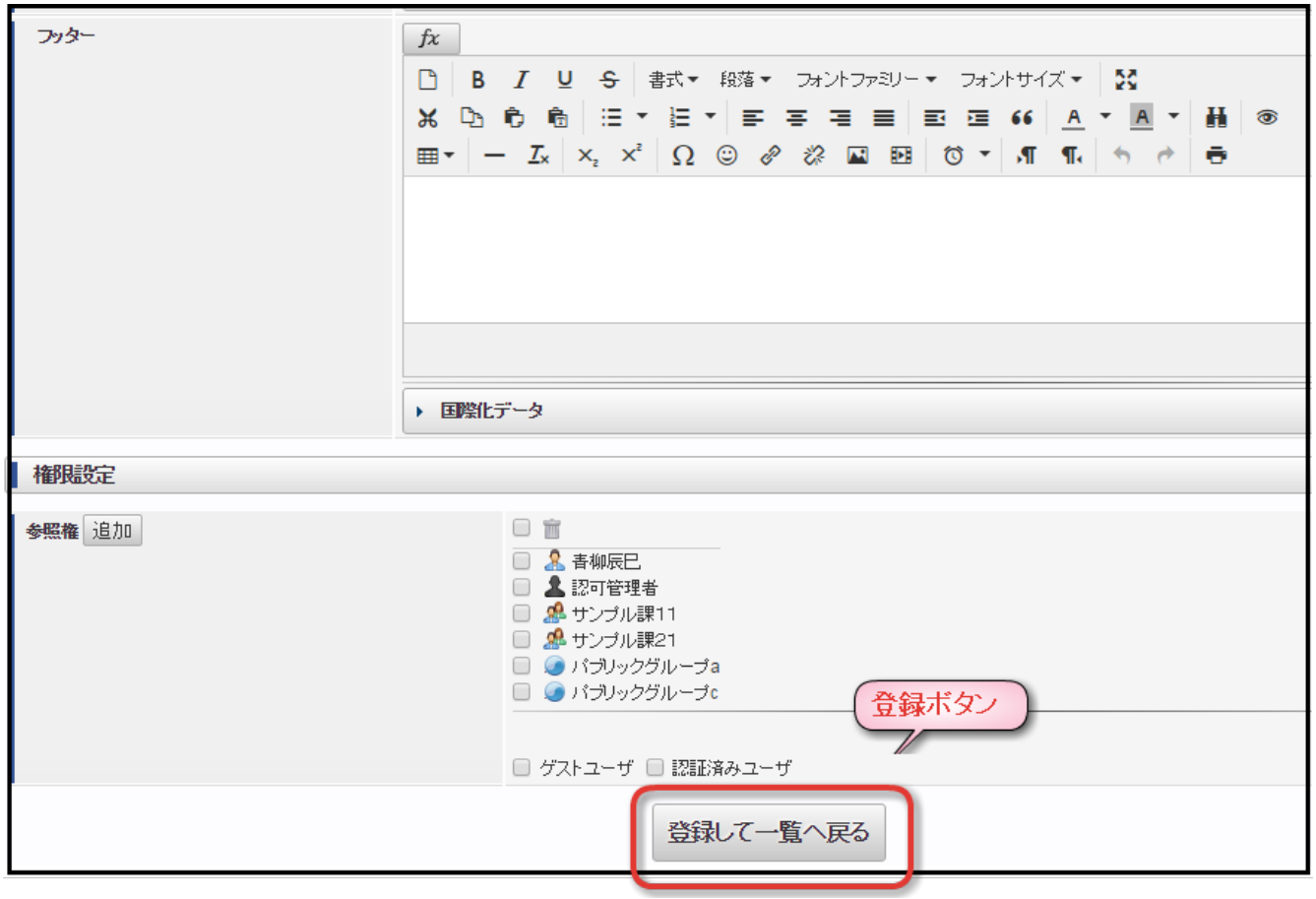
> **コラム**
>
> 参照権の設定は2014 Spring(Granada)から利用可能です。



> **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。

> **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。



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。

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。

応用設定

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。

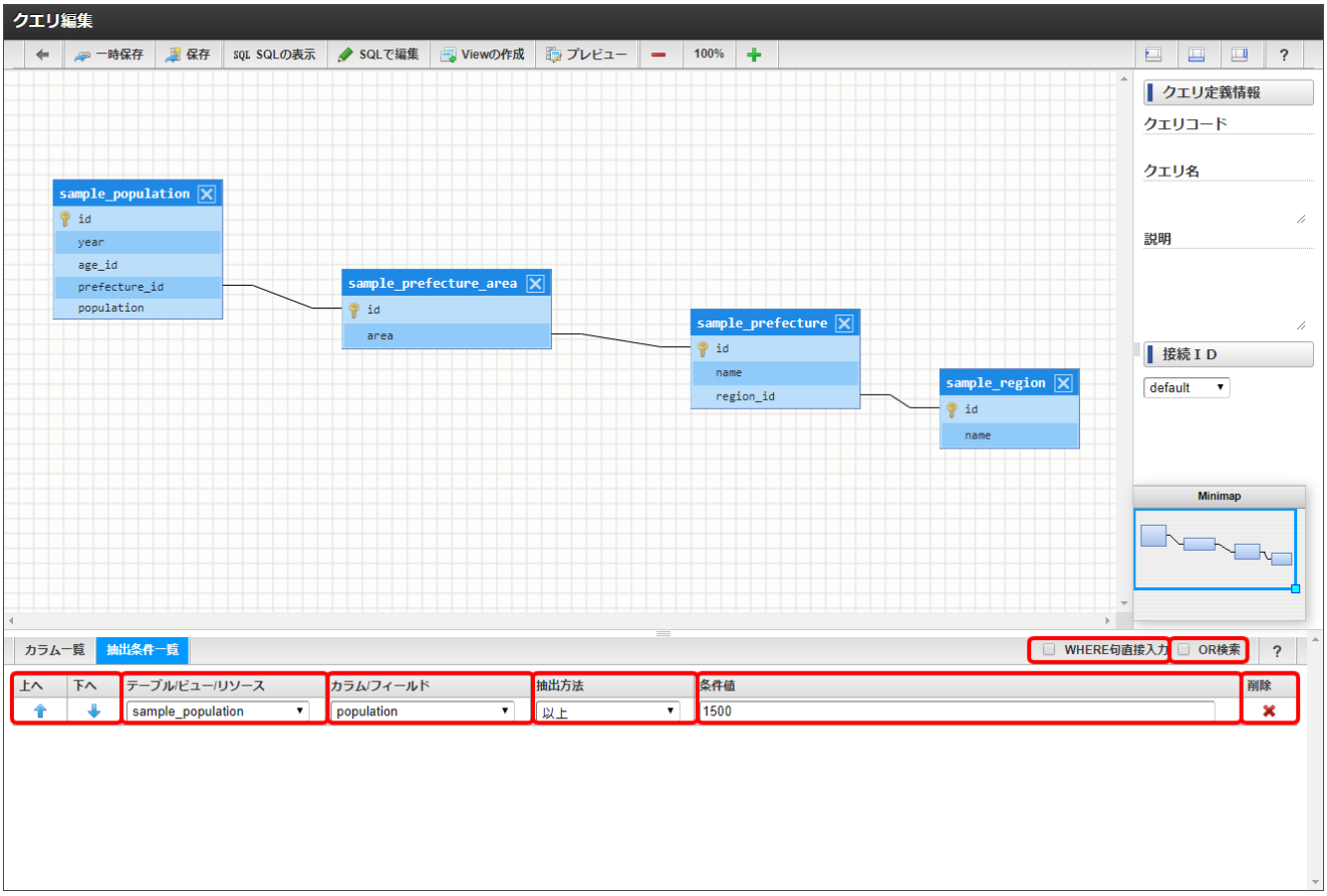
リスト集計の「検索設定」は、データ参照を閲覧するユーザが目的の情報に素早く辿り着くための機能ですが
それに対して「抽出条件」は、ユーザに見せたくない情報を設定者側であらかじめ絞っておくための機能です。

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
  - 動的パラメータを利用可能な個所

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	デザイナ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。
直接入力が選択されると、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。

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。

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

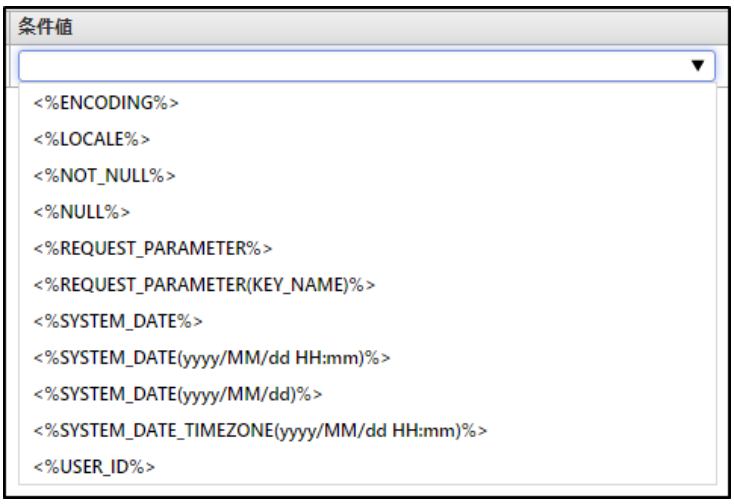
データ参照が閲覧されたときに発行されるSQLは次のような構造です。

```
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]
```

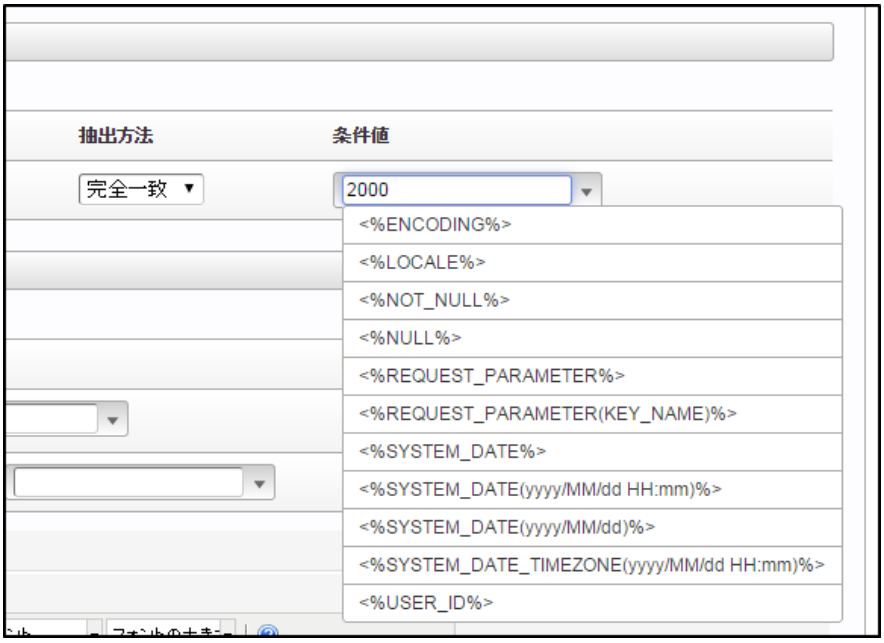
データベースの実装に依存しますが、一般的にはクエリの段階でデータを絞っておいた方がSQLの実行速度は良くなる傾向にあります。
ただし、１つのクエリを共有して複数のデータ参照を作成し、抽出条件はデータ参照ごとに設定した方が良いケースもあるので、適宜使い分けて設定してください。

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。



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、jp.co.intra_mart.foundation.viewcreator.model.Query#execute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。

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。

コラム

- <%REQUEST_PARAMETER(KEY_NAME)%>の使用例

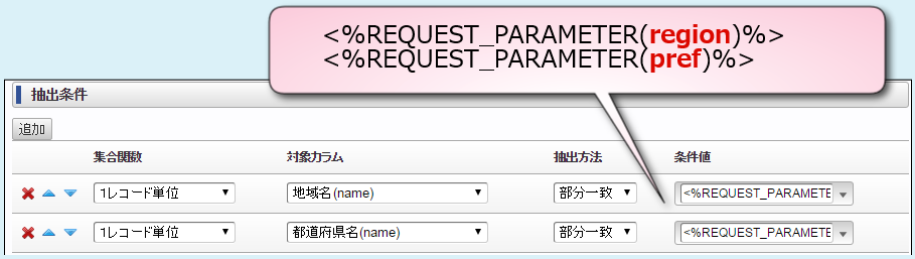
1. 検索設定の場合

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。

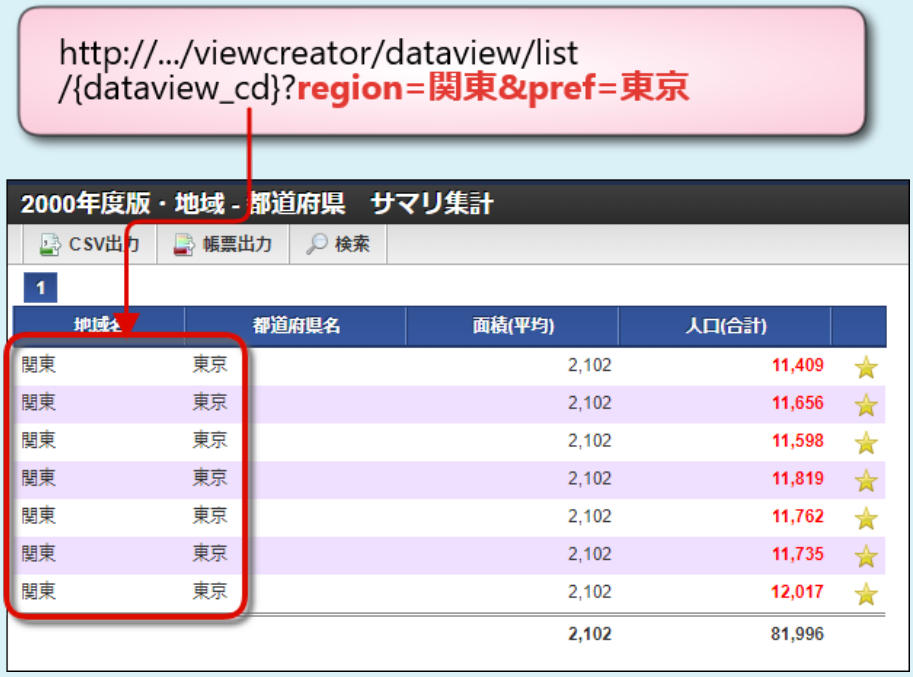
データ参照を表示した際に指定したリクエストパラメータを受け取ることができます。

2. 抽出条件の場合

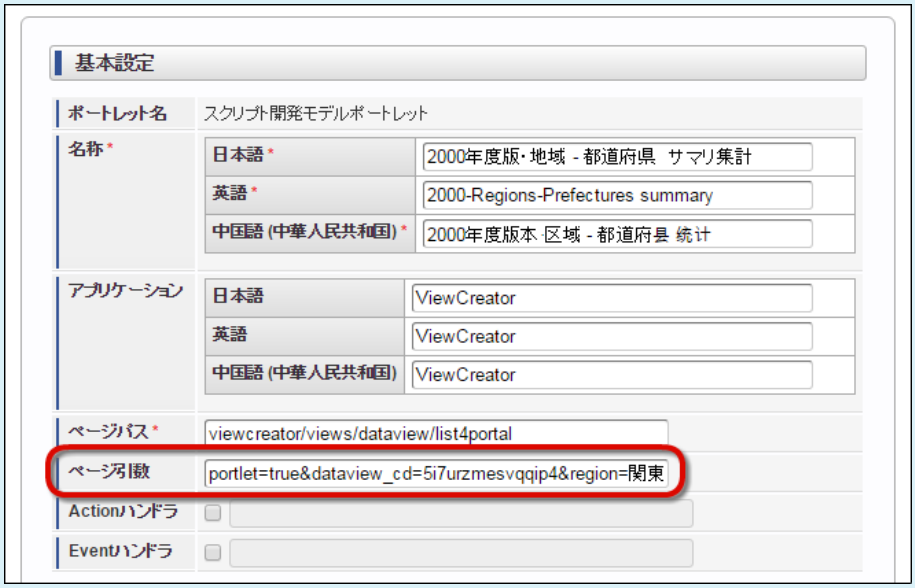
タグが受け取る引数として、リクエストパラメータのキー名を任意に指定します。



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。



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。



「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。

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。

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。

動的パラメータを利用可能な個所

動的パラメータは「抽出条件」や「検索設定」の他にも、SQLを記述する設定項目において利用可能な場合があります。
詳細は以下の通りです。

定義の種類	画面・設定項目		データの扱い
クエリ	クエリビルダ	抽出条件（抽出方法が直接入力のとき）	そのまま利用
		抽出条件（抽出方法が直接入力以外のとき）	自動変換

定義の種類	画面・設定項目		データの扱い
		WHERE句直接入力	そのまま利用
		Viewの作成	利用不可
	SQLビルダ	SQL	そのまま利用
データ参照		抽出条件	自動変換
		検索設定	自動変換

「データの扱い」が「そのまま利用」の場合は、動的パラメータから取得したデータを加工無しでSQL文の一部として利用するため、そのことを考慮した記述にする必要があります。
たとえば、動的パラメータで取得した値を文字列として利用する場合は、シングルクォートで囲む必要があります。
「自動変換」の場合は、データ型に応じたSQL文に変換してから利用するため、そのような考慮は不要です。

コラム

使用例

以下は「*SQLビルダによるクエリの作成*」の「SQL」で利用する場合の記述例です。
imm_user テーブルからレコードを取得するときの条件として user_cd に指定する値を
<%REQUEST_PARAMETER(KEY_NAME)%> から取得します。
このとき、<%REQUEST_PARAMETER(KEY_NAME)%> 全体をシングルクォーテーションで囲っています。

```
SELECT
  *
FROM
  imm_user
WHERE
  user_cd = '<%REQUEST_PARAMETER(userCd)%>'
```

コラム

<%REQUEST_PARAMETER(KEY_NAME)%> で受け取ったデータの中に ‘（シングルクォーテーション）が含まれている場合、’’（シングルクォーテーションを２つ連続して記述）に変換されます。

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。

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

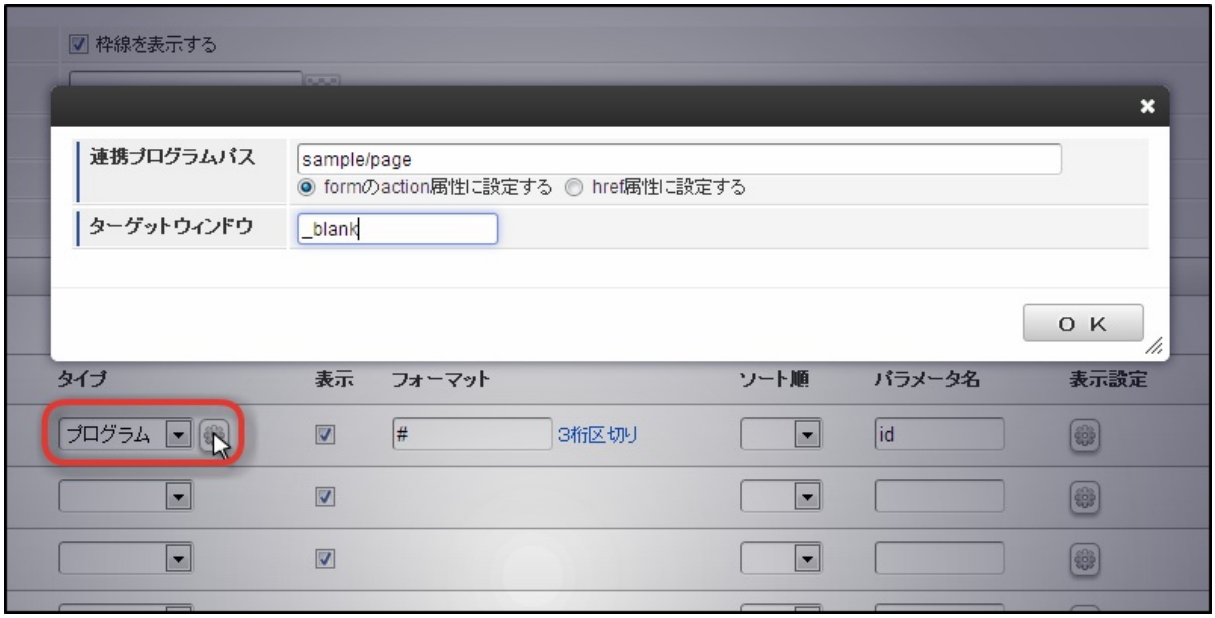
- 設定方法

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。



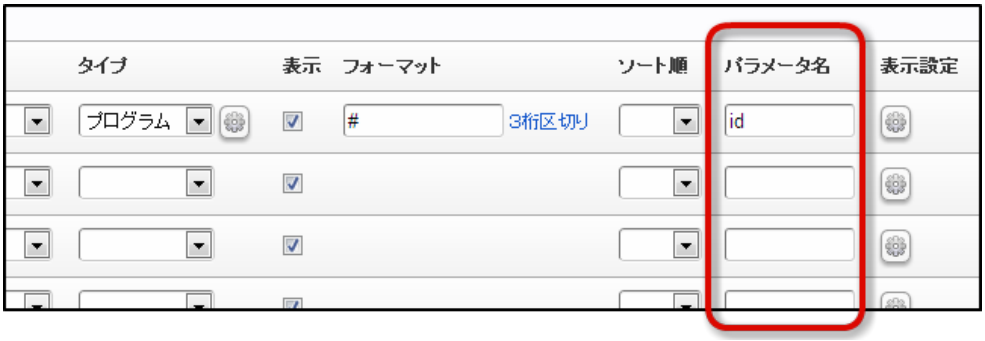
リンクがクリックされると、連携プログラムパスで設定されたプログラムへ連携されます。

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

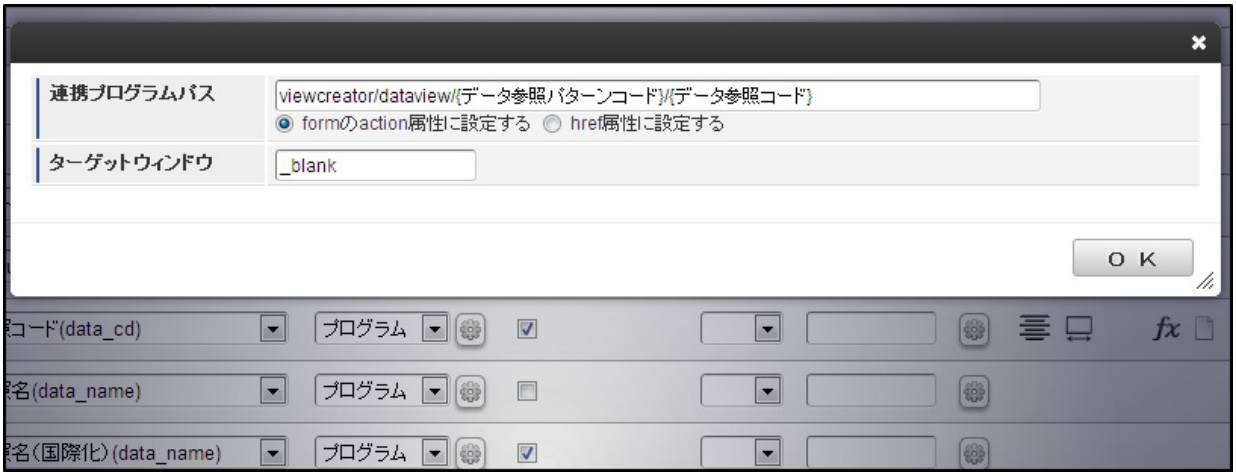
データ型	説明
文字列型	カラムに格納されているデータをそのまま送信します。
数値型	カラムに格納されているデータの文字列表現を送信します。 フォーマット文字列の適用は行いません。
日付型	ミリ秒に変換された数値文字列データと、フォーマット文字列を適用したデータを送信します。 上記のデータは別々のパラメータとして送信され、後者のパラメータのキー名には 「_format」が付加されます。 フォーマット文字列はデータ参照の定義に含まれるカラムの「フォーマット」の設定値が適用されます。 フォーマット文字列が未設定または空文字の場合は、JavaScriptのDateオブジェクトの文字列表現が送信されま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

PathVariablesへの対応

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。

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。

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

適宜、適切なアクセス権を設定します。

登録したポートレット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。

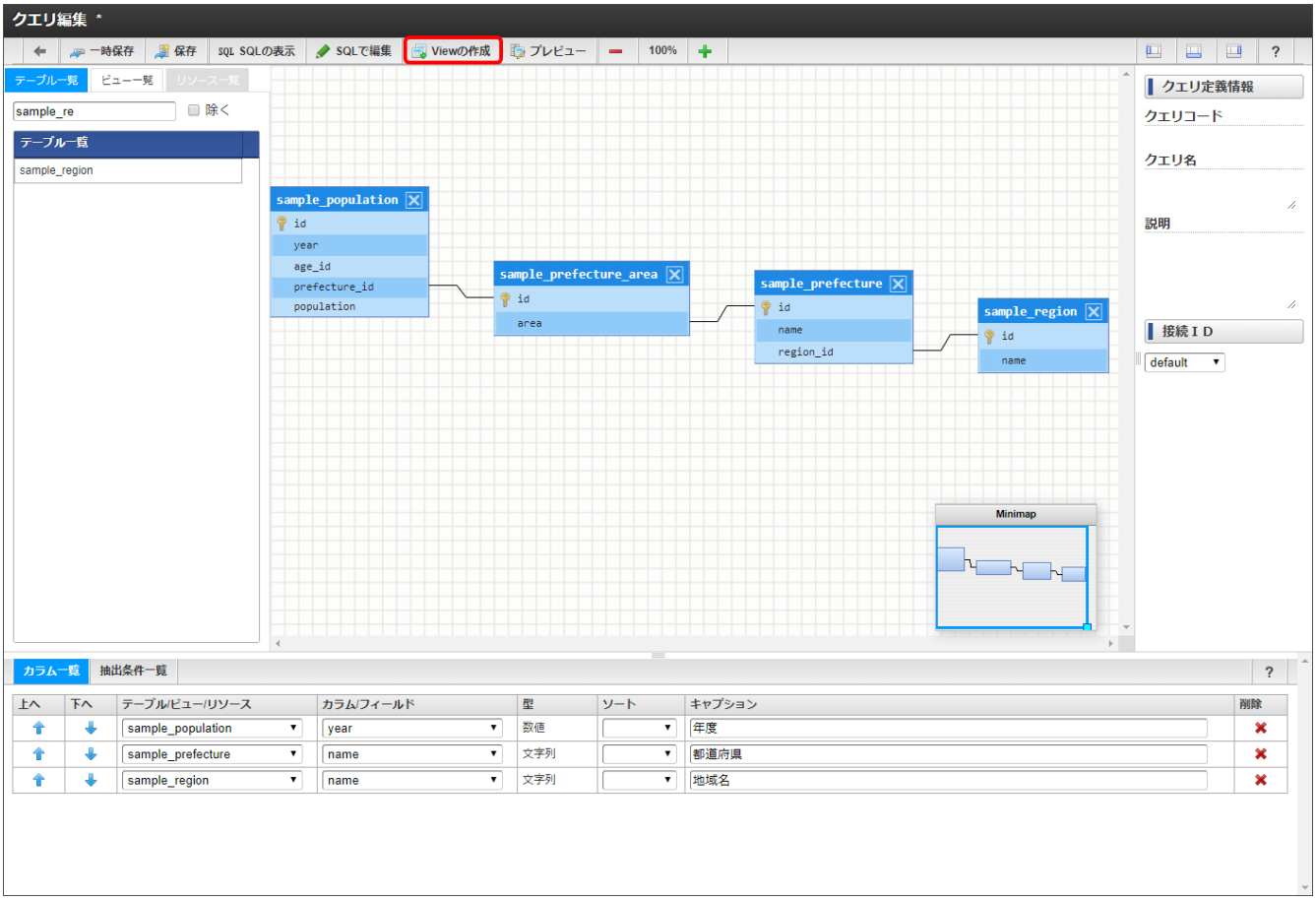
物理Viewの作成と表示

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。

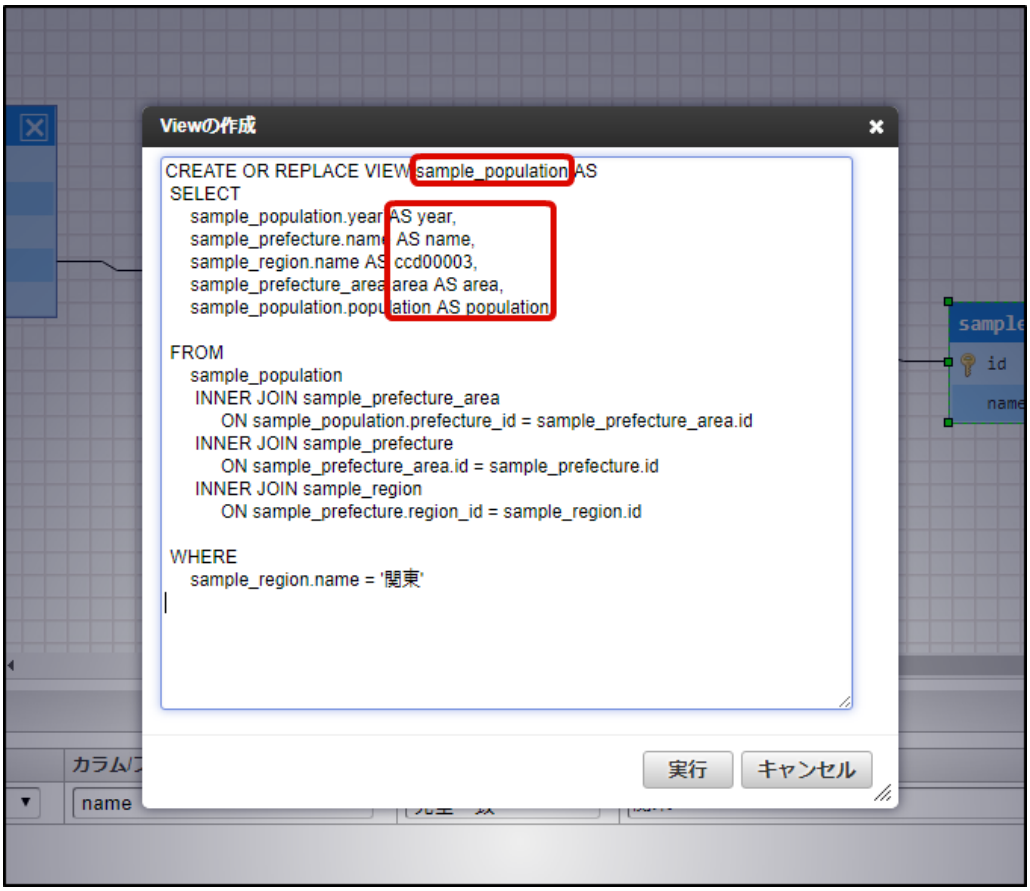
※ViewCreator機能の中でのデータ参照（View）のことではありません。

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。

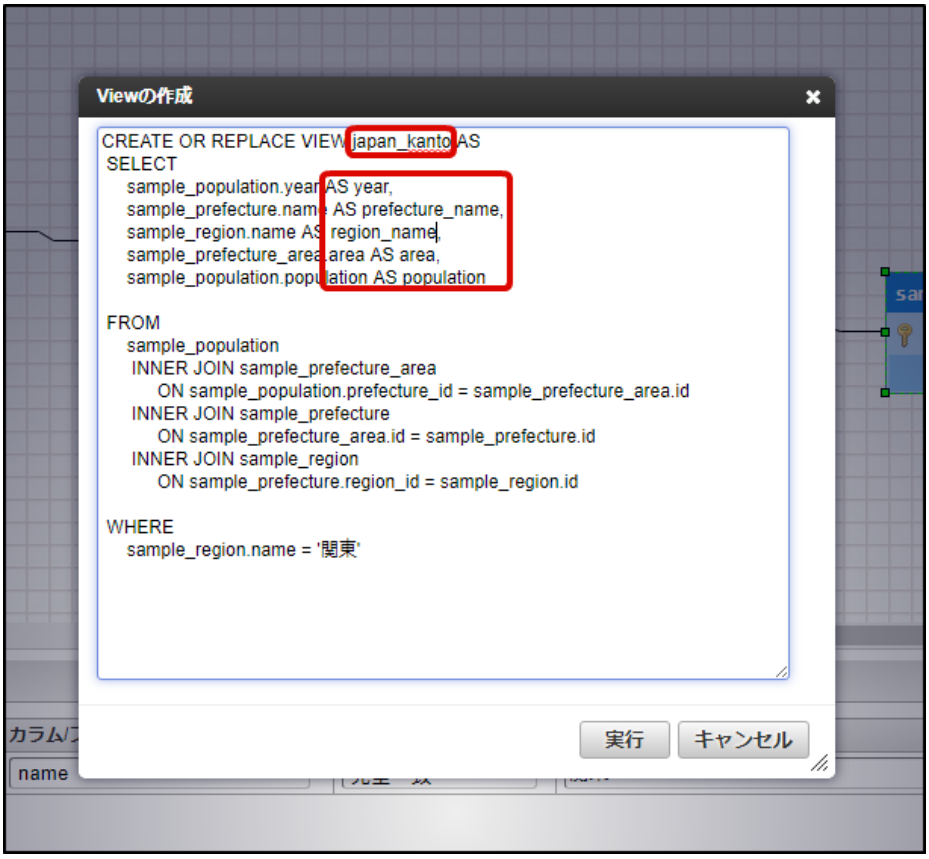
クエリ編集画面のツールバーから、「Viewの作成」をクリックします。



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。

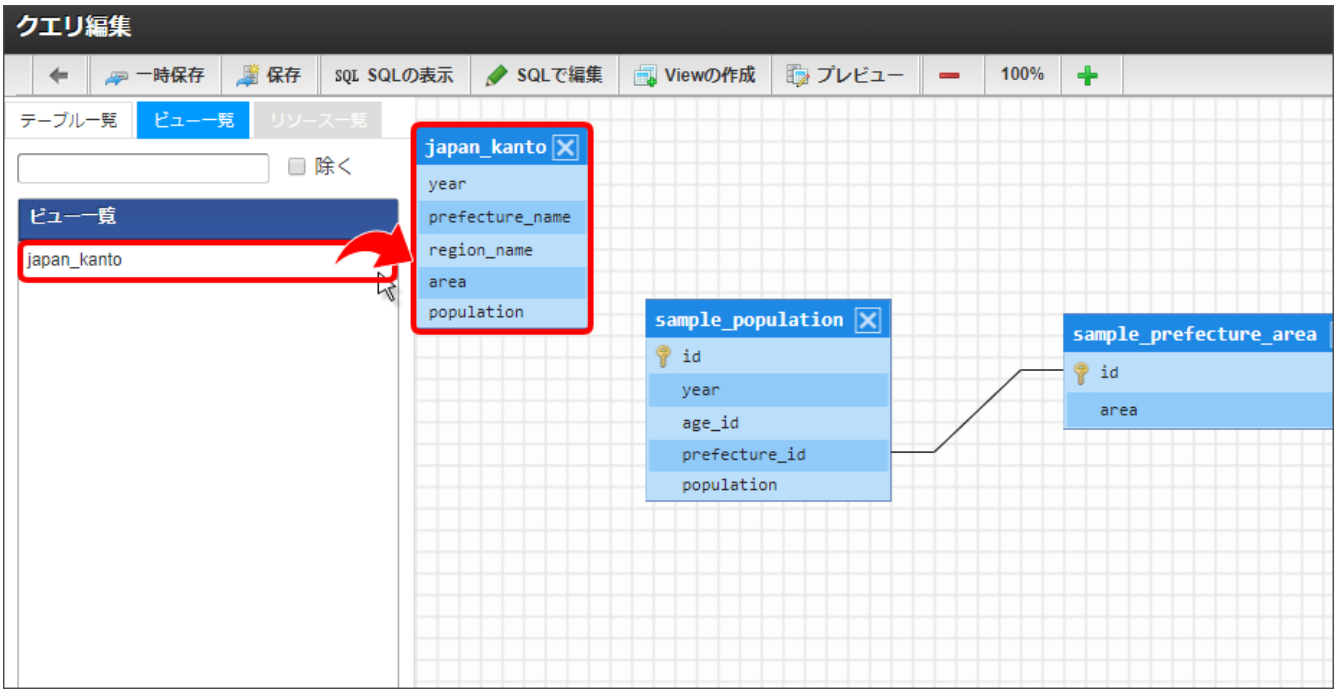


Viewとフィールドの名称を確認し、適宜SQL文を編集してください。



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。

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

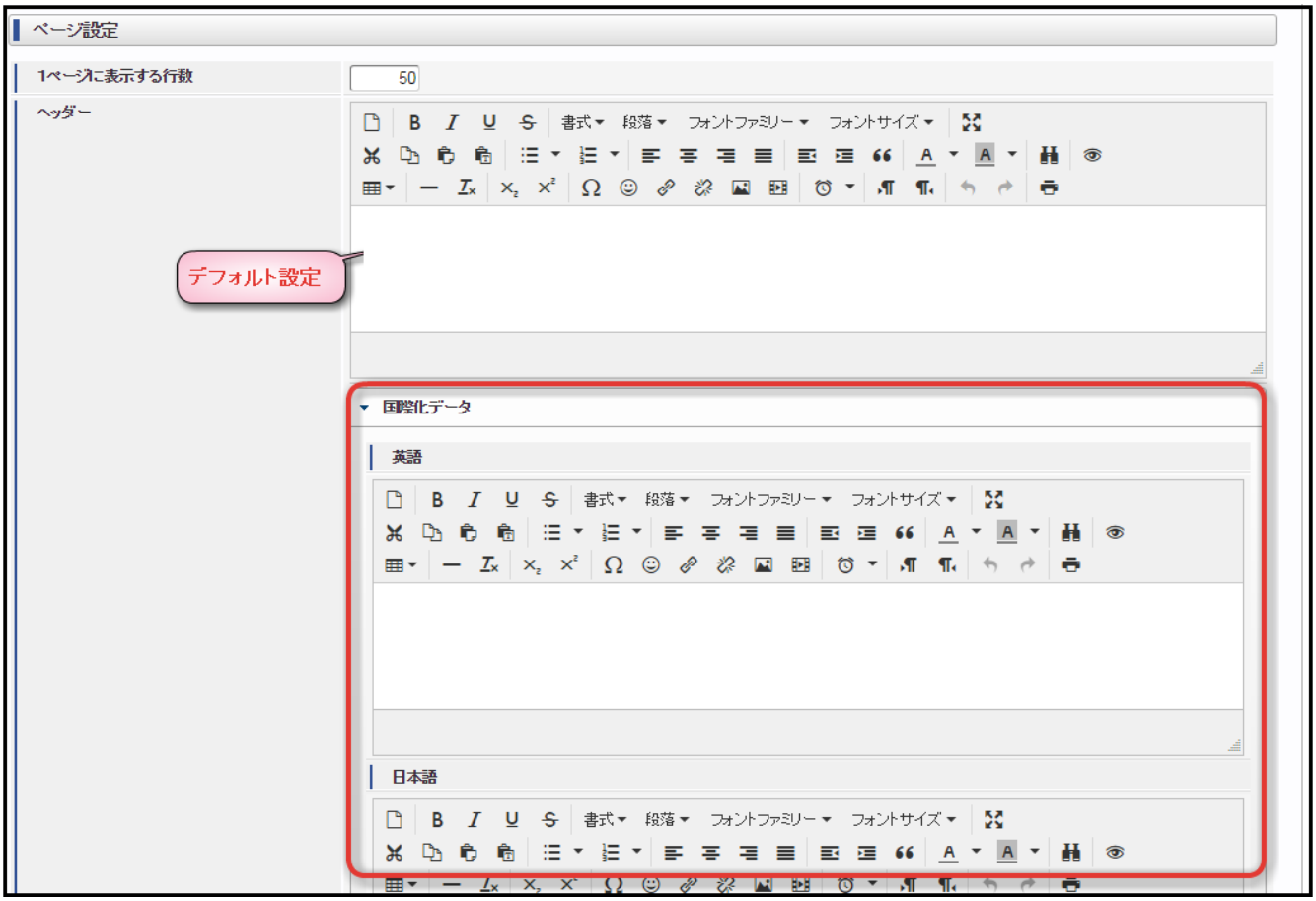
データ参照に付加的に説明などを表示したい場合には、ヘッダおよびフッタの設定を利用してください。
また、ヘッダ・フッタとデータ参照名は言語ごとに表示を変えることが可能です。

- 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。

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。

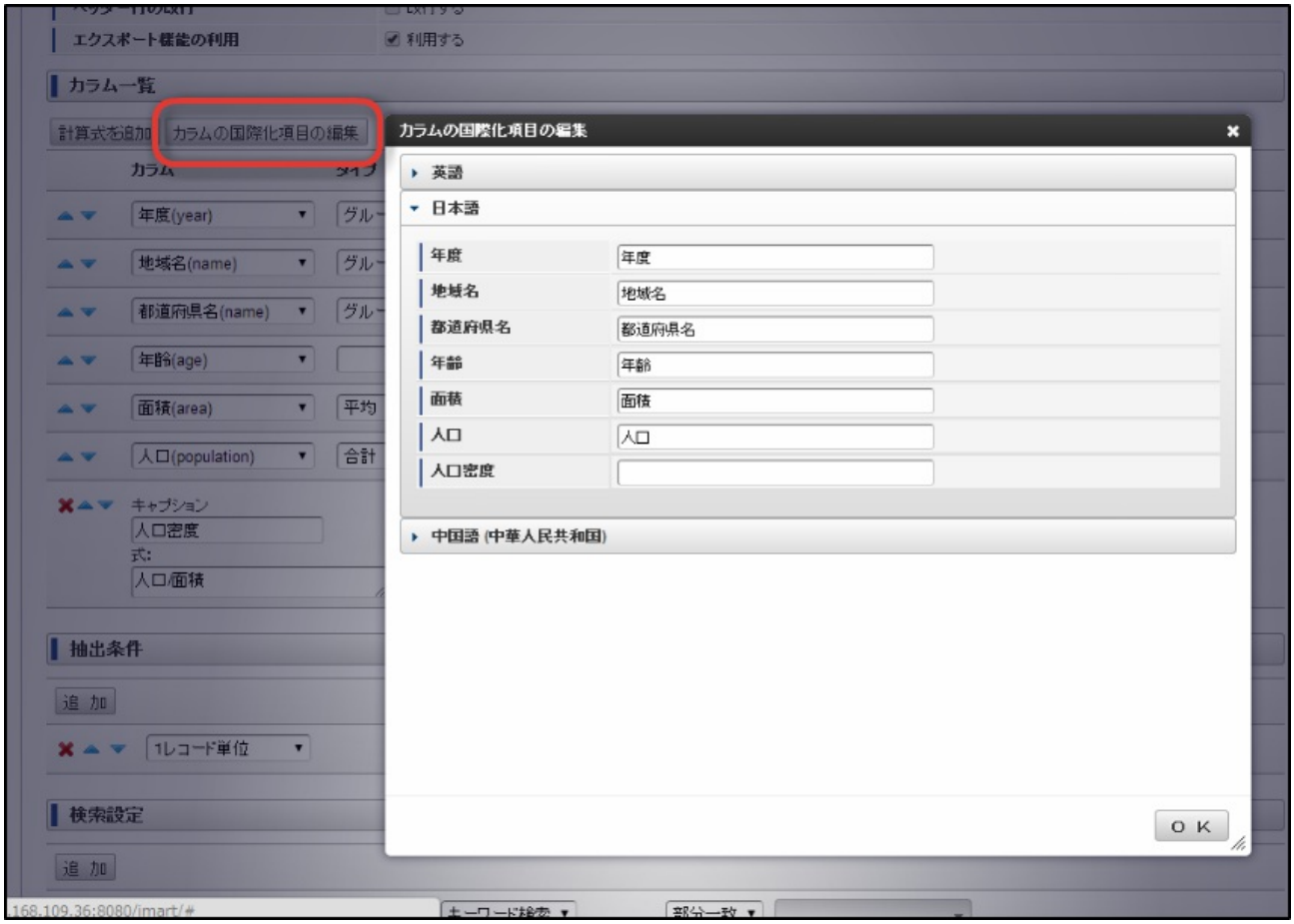
ヘッダとフッタも同様に言語ごとに設定を変えることができます。



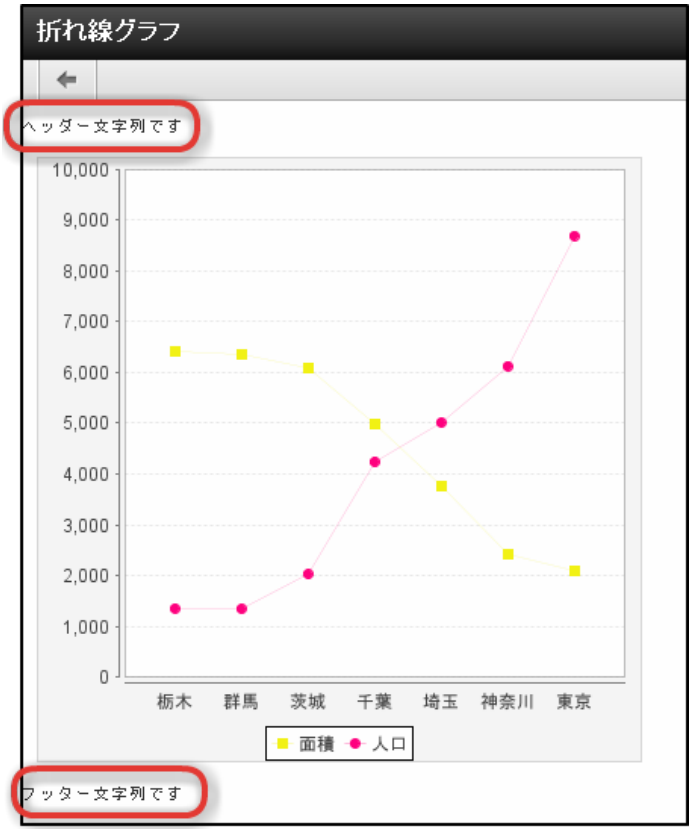
ヘッダとフッタはリッチテキストが利用可能です。

> **コラム**
>
> リッチテキストは2014 Spring(Granada)から利用可能です。

カラムヘッダの表示名については、「カ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。



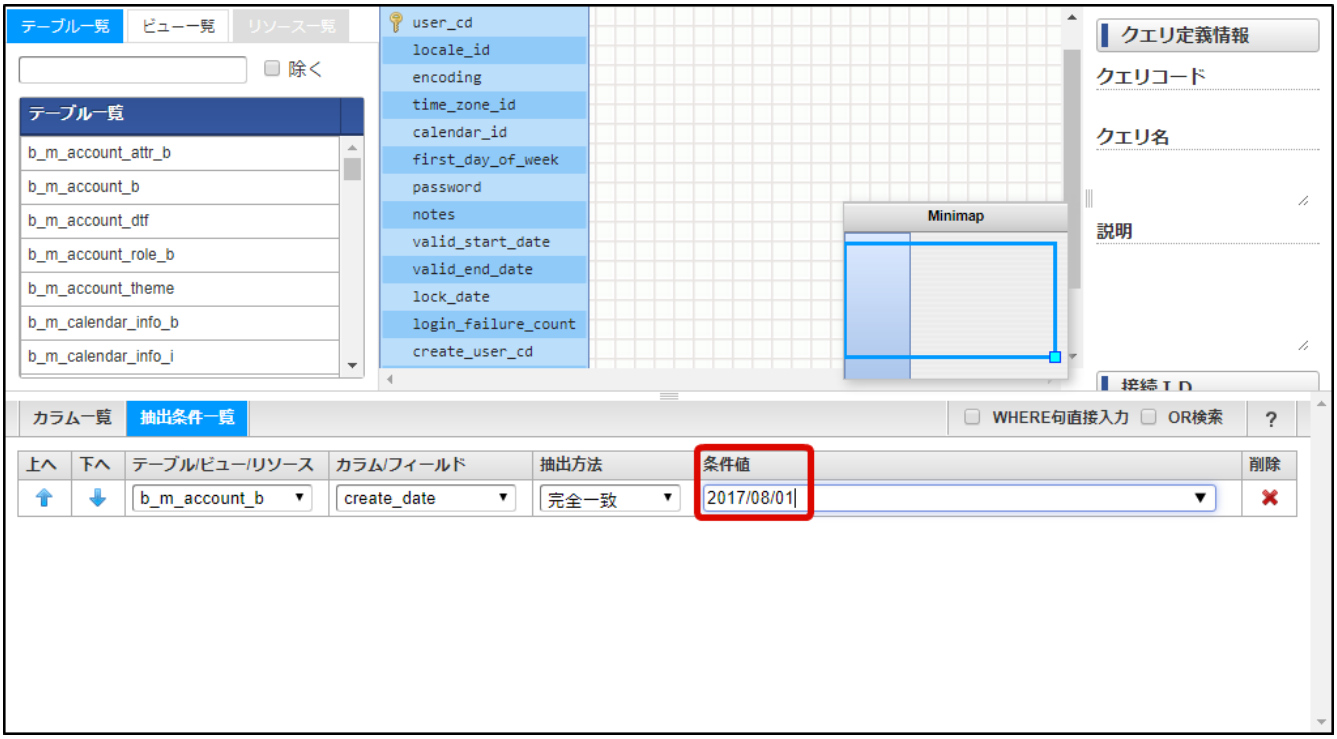
- 表示例



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

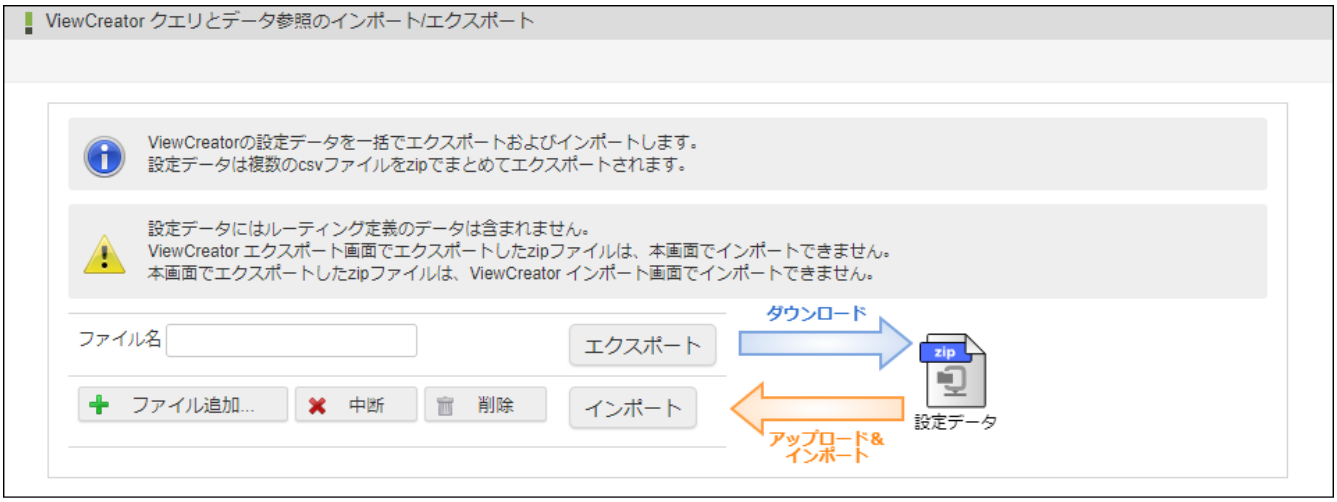
- クエリとデータ参照のインポート/エクスポート
- インポート
  - インポート画面を確認する
  - インポート種別を選択する
  - インポートを実行する
  - インポート結果を確認する
- エクスポート
  - エクスポート画面を確認する
  - エクスポート対象を選択する
  - エクスポートを実行する

クエリとデータ参照のインポート/エクスポート

注意

2021 Winter(Dandelion) から本機能は**非推奨**となりました。
クエリやデータ参照のインポートおよびエクスポートを行う場合は「*インポート*」、「*エクスポート*」を使用してください。

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。



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。

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で作成したクエリやデータ参照および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。

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。

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。

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アコーディオン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

スキップ件数が1件以上の場合

スキップファイル詳細のアコーディオンを押下する

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
27	16	11

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
1	0	1

▼ 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
sample_category.xml	sample_category	サンプルカテゴリ

ルーティング定義

総件数	新規登録件数	スキップ件数
2	1	1

▼ 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
8g6ty2ma032w7.xml	8g6ty2ma032w7	システムログ

クエリ定義

総件数	新規登録件数	スキップ件数
5	2	3

▼ スキップファイル詳細

ファイル名	クエリコード	クエリ名
5i7ur3opljh2x.xml	5i7ur3opljh2x	ViewCreatorの設定テーブルを利用したクエリ - クエリ・データ参照
5ib8yonru57ws.xml	5ib8yonru57ws	system_log
5ib8yony9uhzy.xml	5ib8yony9uhzy	transition_log

データ参照定義

総件数	新規登録件数	スキップ件数
19	13	6

▼ スキップファイル詳細

ファイル名	データ参照コード	データ参照名
5i7ur3pgur5npp4.xml	5i7ur3pgur5npp4	データ参照一覧・サンプル2(クエリ別でグループ化)
5ib8yonzvfh119v.xml	5ib8yonzvfh119v	トランジションログ・ページ別グループ化リスト
5ib8yooamstrf9v.xml	5ib8yooamstrf9v	[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
5ib8yoocovroj9v.xml	5ib8yoocovroj9v	エラーログ
5ib8yot4emfmr9v.xml	5ib8yot4emfmr9v	システムログ
dataview_list_sample.xml	dataview_list_sample	データ参照一覧・サンプル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。

エクスポート

「ViewCreator エクスポート」画面を利用すると、ViewCreatorで作成したクエリやデータ参照および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。

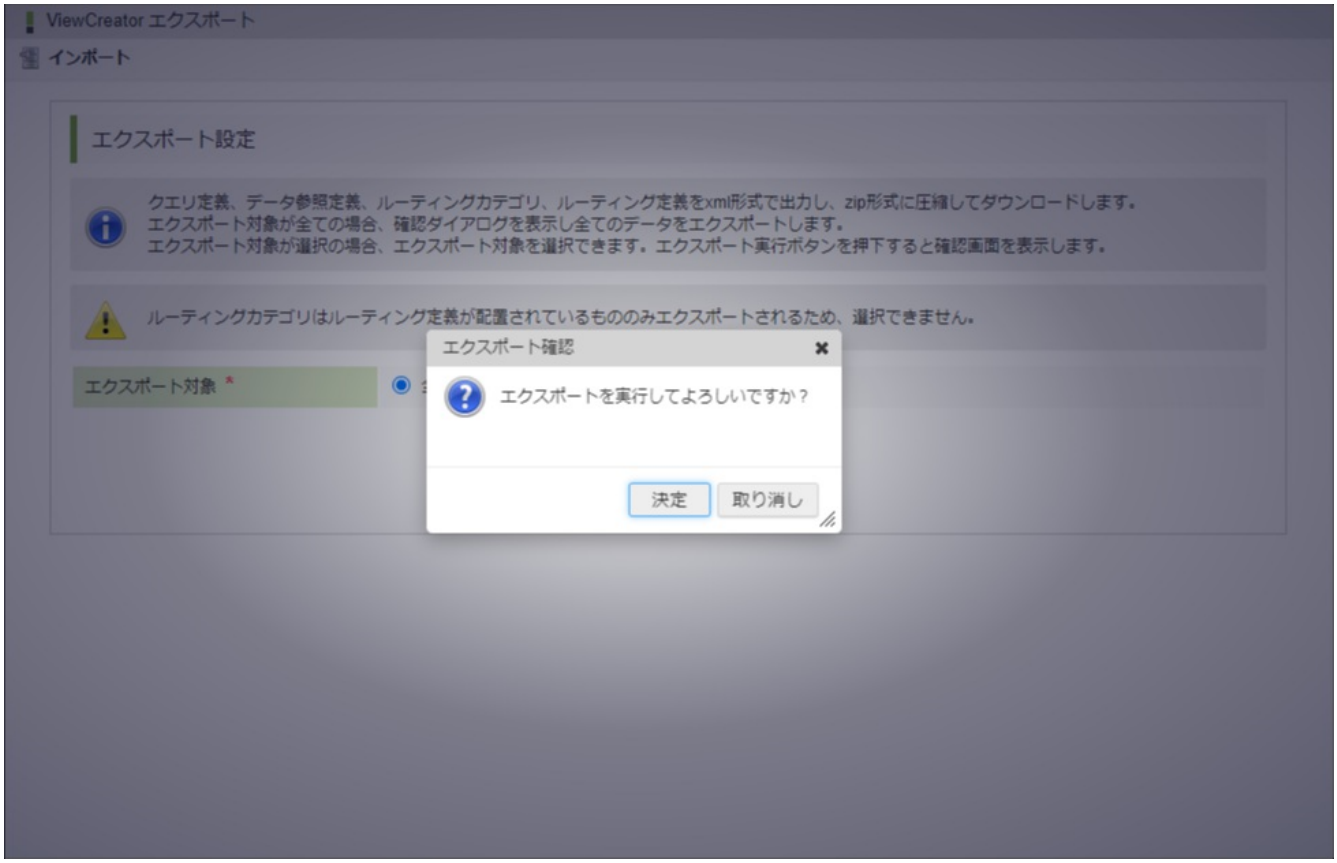
エクスポートを実行する

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。

エクスポート設定

クエリ定義、データ参照定義、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

☐	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
☐	🔍	sample_category	サンプルカテゴリ	8g6ty5f4i32w8	エラーログ	5ib8yonru57ws	system_log
☐	🔍	sample_category	サンプルカテゴリ	8g6ty2ma032w7	システムログ	5ib8yonru57ws	system_log

1ページ中 1 ページ目 50 2 件中 1 - 2 を表示

クエリ定義

選択済み件数 : **0 件**

クエリ名検索

☐	参照	クエリコード	クエリ名	説明
☐	🔍	5i7o4o3kli1pj	日本のデータ	
☐	🔍	5i7ur3opljh2x	ViewCreatorの設定テーブルを利用したクエリ - クエリ・データ参照	
☐	🔍	5ib8yonru57ws	system_log	
☐	🔍	5ib8yony9uhzy	transition_log	
☐	🔍	8fhl6zaswe6qchu	日本のデータ（データ結合）	

1ページ中 1 ページ目 50 5 件中 1 - 5 を表示

データ参照定義

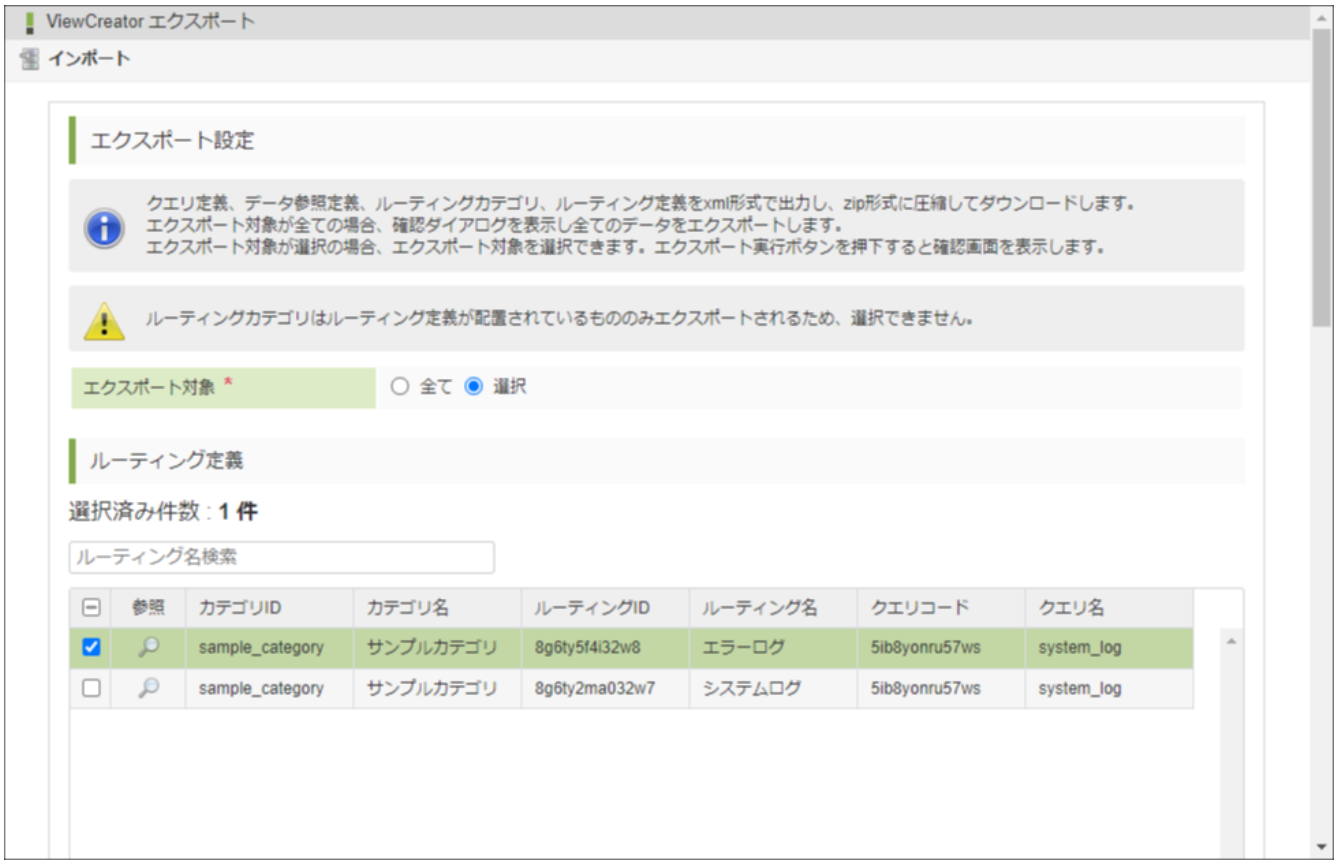
選択済み件数 : **0 件**

データ参照名検索

☐	参照	データ参照コード	データ参照名	集計パターン	説明
☐	🔍	5ib8yooamstrf9v	[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	リスト	
☐	🔍	5i7urzmesvqqip4	2000年度版・地域 - 都道府県サマリ集計	サマリ	
☐	🔍	5i7urzmlbc9svp4	2000年度版・地域 - 都道府県サマリ集計/面積順	サマリ	
☐	🔍	8fhjsbw2gwvy4hu	2010年度版・地域 - 都道府県サマリ集計	サマリ	
☐	🔍	8fhjsvr2fwzi5hu	2010年度版・地域 - 都道府県サマリ集計/面積順	サマリ	

1ページ中 1 ページ目 50 19 件中 1 - 19 を表示

2. 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。

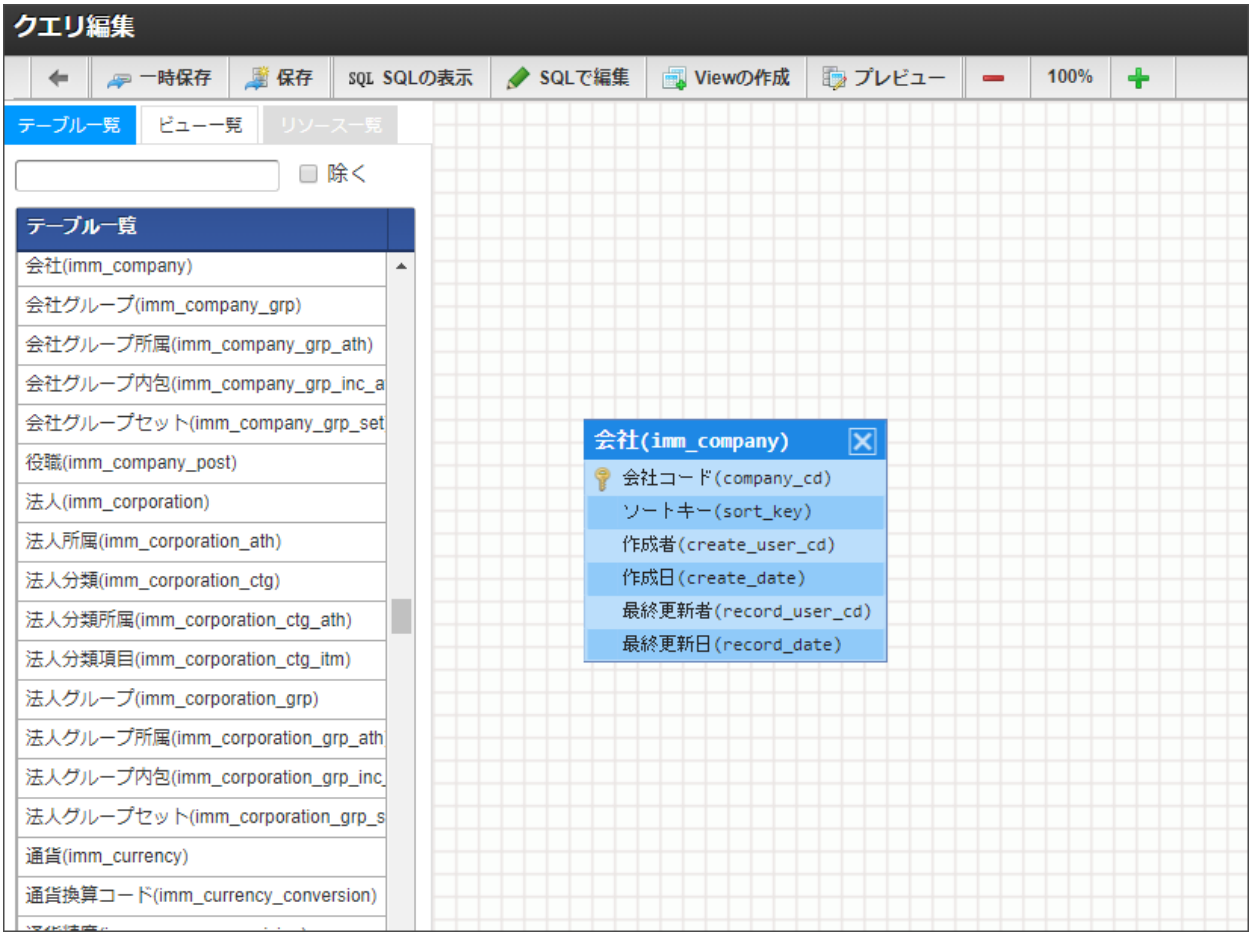


3. 「ViewCreator エクスポート対象確認」画面が表示されるので、エクスポート対象を確認し「決定」をクリックします。

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。



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。

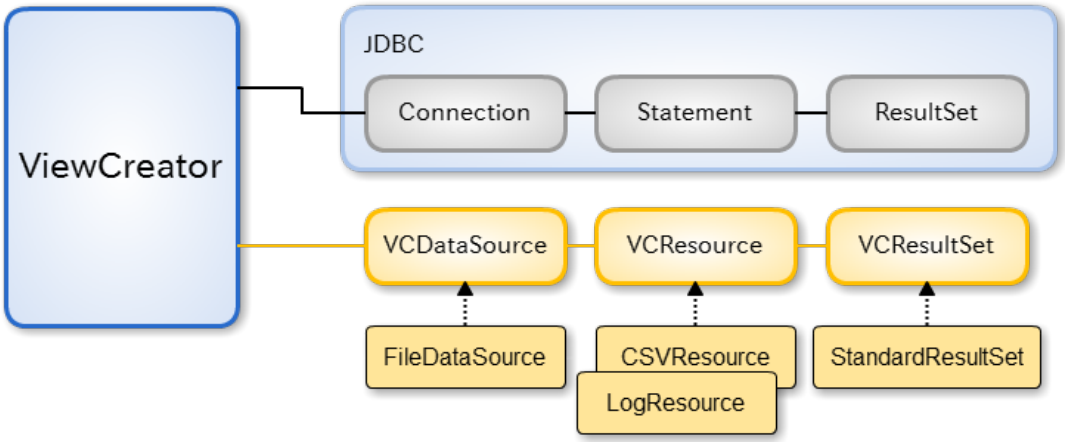
外部データソースの利用

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



- **VCDataSourceインタフェース**

このインタフェースの実装は、複数のVCResource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。

- **VCResourceインタフェース**

VCResource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**

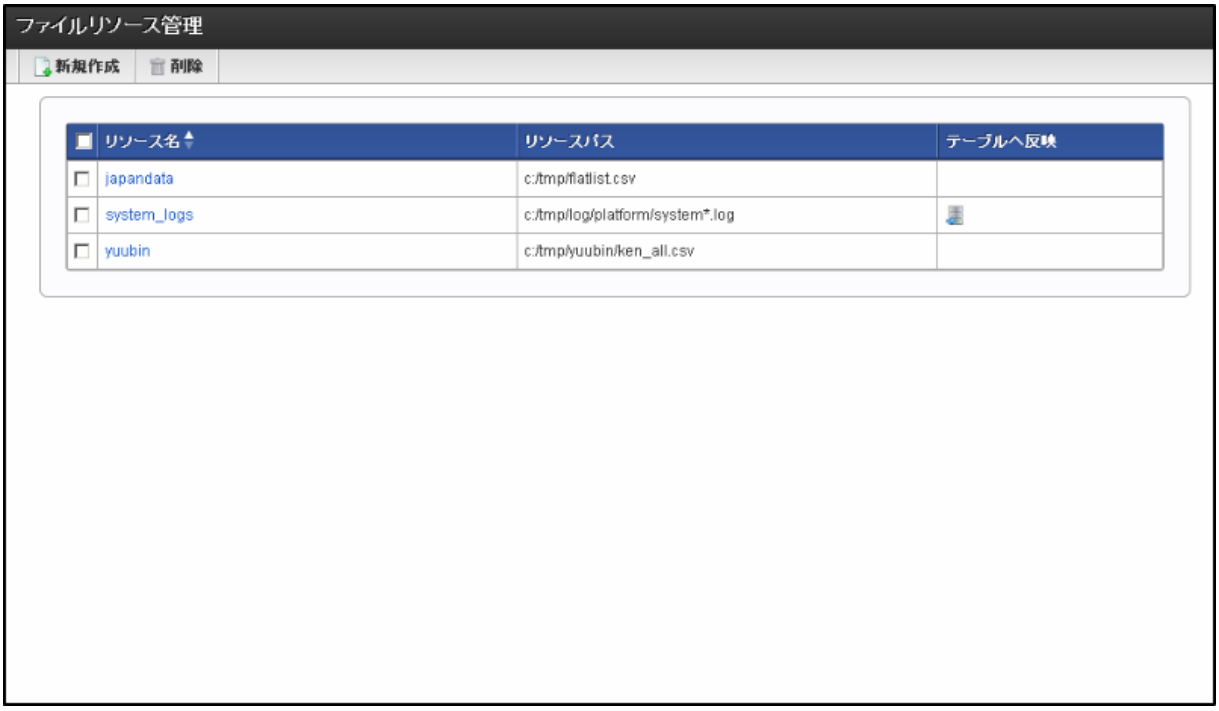
リソースから取得した結果データセットを表します。

> 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.VCDataSource
> jp.co.intra_mart.foundation.viewcreator.expansion.VCResource
> jp.co.intra_mart.foundation.viewcreator.expansion.VCResultSet

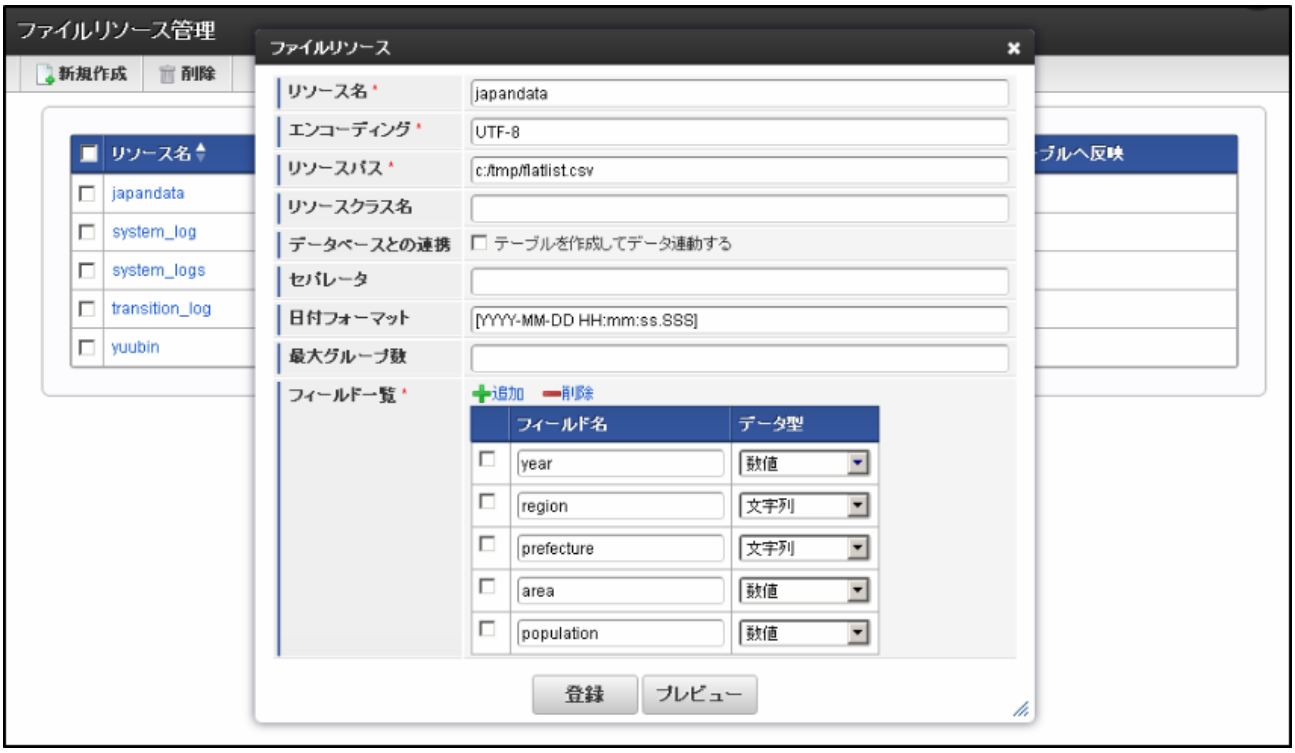
CSVファイルおよびログファイルのリソース登録

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	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。 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。 例）system/platform/system*.log 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。 デフォルトでは相対パスのみ入力が許可されています。
リソースクラス名	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。 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。 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。
データベースとの連携	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。 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。 ※テーブル名は、「imvc_rc_」 + リソース名です。 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。 jp.co.intra_mart.foundation.viewcreator.expansion.impl.ResourceWithTable ※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。
セパレータ	データの区切り文字を指定します。 設定が省略された場合、 CSVResourceクラスのデフォルト値は、「,（カンマ）」 LogResourceクラスのデフォルト値は、「\t（TAB）」です。
日付フォーマット	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。
最大グループ数	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。 OutOfMemoryErrorを起こさないようにするための設定です。 省略時のデフォルト値は20000です。
フィールド一覧	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。 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。 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す。

コラム

リソースパスには「ファイルリソースに設定可能なパス形式」を変更することで、%LOG_DIRECTORY% および絶対パスが設定できます。
%LOG_DIRECTORY% は intra-mart Accel Platform が出力するログファイルの出力先に変換されます。

ログファイルの出力先については以下を参照してください。
「コアモジュール-サーバコンテキスト設定」

「ファイルリソースに設定可能なパス形式」 は 2023 Autumn(Hollyhock)から利用可能です。

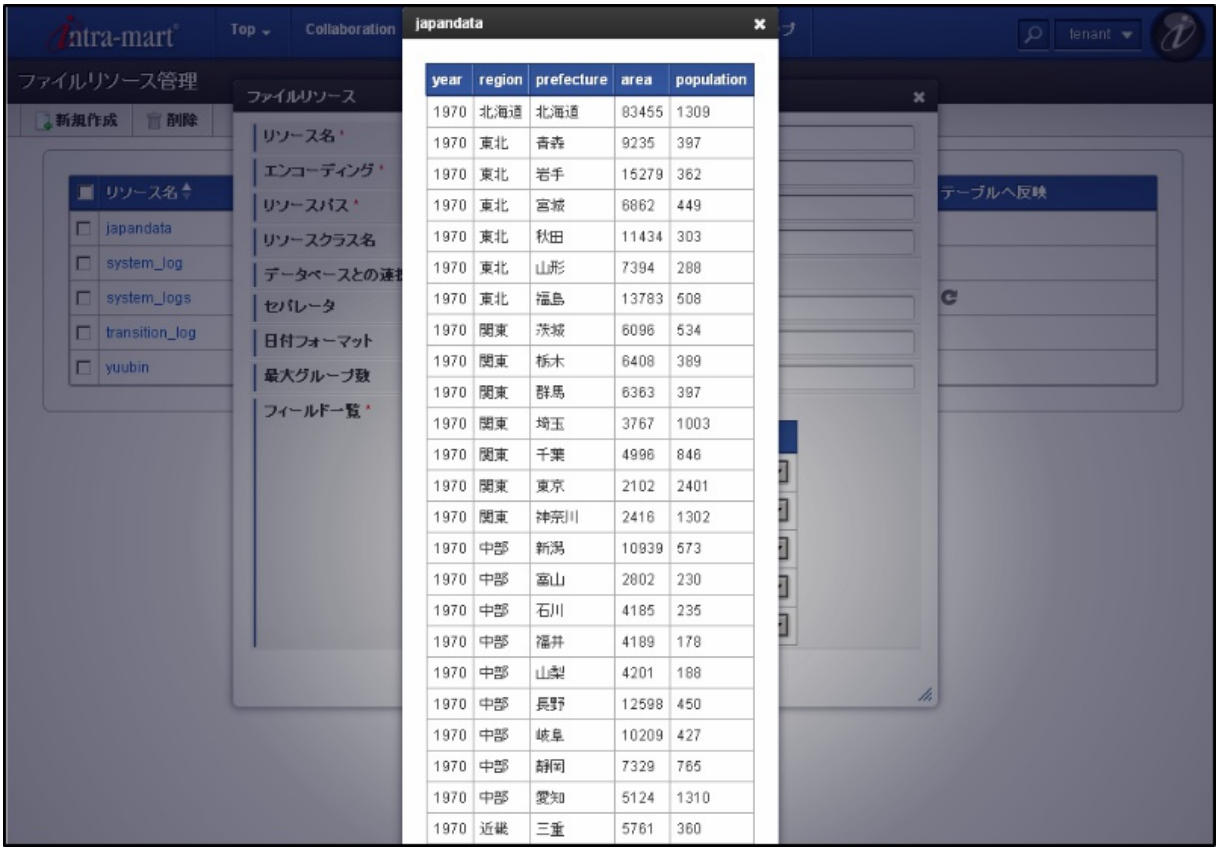
注意

%LOG_DIRECTORY% および絶対パスは分散環境に対応していません。
分散環境で利用する場合、想定していないファイルが取得されたり、ファイルが取得できないことがあります。

注意

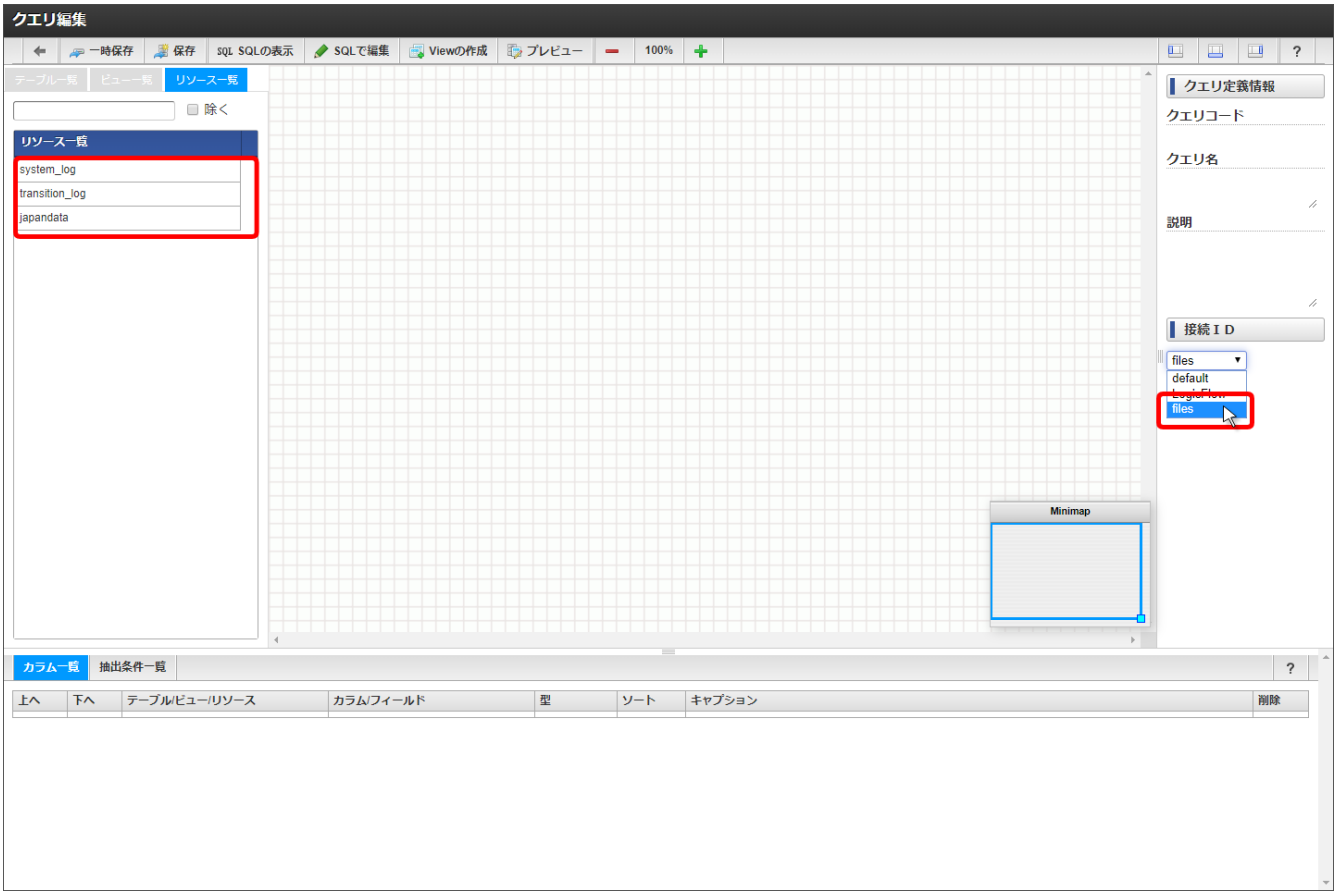
標準のCSVResourceクラスとLogResourceクラスはソートの実装を提供しません。
従って、これらの標準クラスを使用して作成されたリスト集計ではソートに関する全ての機能を利用できません。

「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。



year	region	prefecture	area	population
1970	北海道	北海道	83455	1309
1970	東北	青森	9235	397
1970	東北	岩手	15279	362
1970	東北	宮城	6862	449
1970	東北	秋田	11434	303
1970	東北	山形	7394	288
1970	東北	福島	13783	508
1970	関東	茨城	6096	534
1970	関東	栃木	6408	389
1970	関東	群馬	6363	397
1970	関東	埼玉	3767	1003
1970	関東	千葉	4996	846
1970	関東	東京	2102	2401
1970	関東	神奈川	2416	1302
1970	中部	新潟	10939	573
1970	中部	富山	2802	230
1970	中部	石川	4185	235
1970	中部	福井	4189	178
1970	中部	山梨	4201	188
1970	中部	長野	12598	450
1970	中部	岐阜	10209	427
1970	中部	静岡	7329	765
1970	中部	愛知	5124	1310
1970	近畿	三重	5761	360

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。

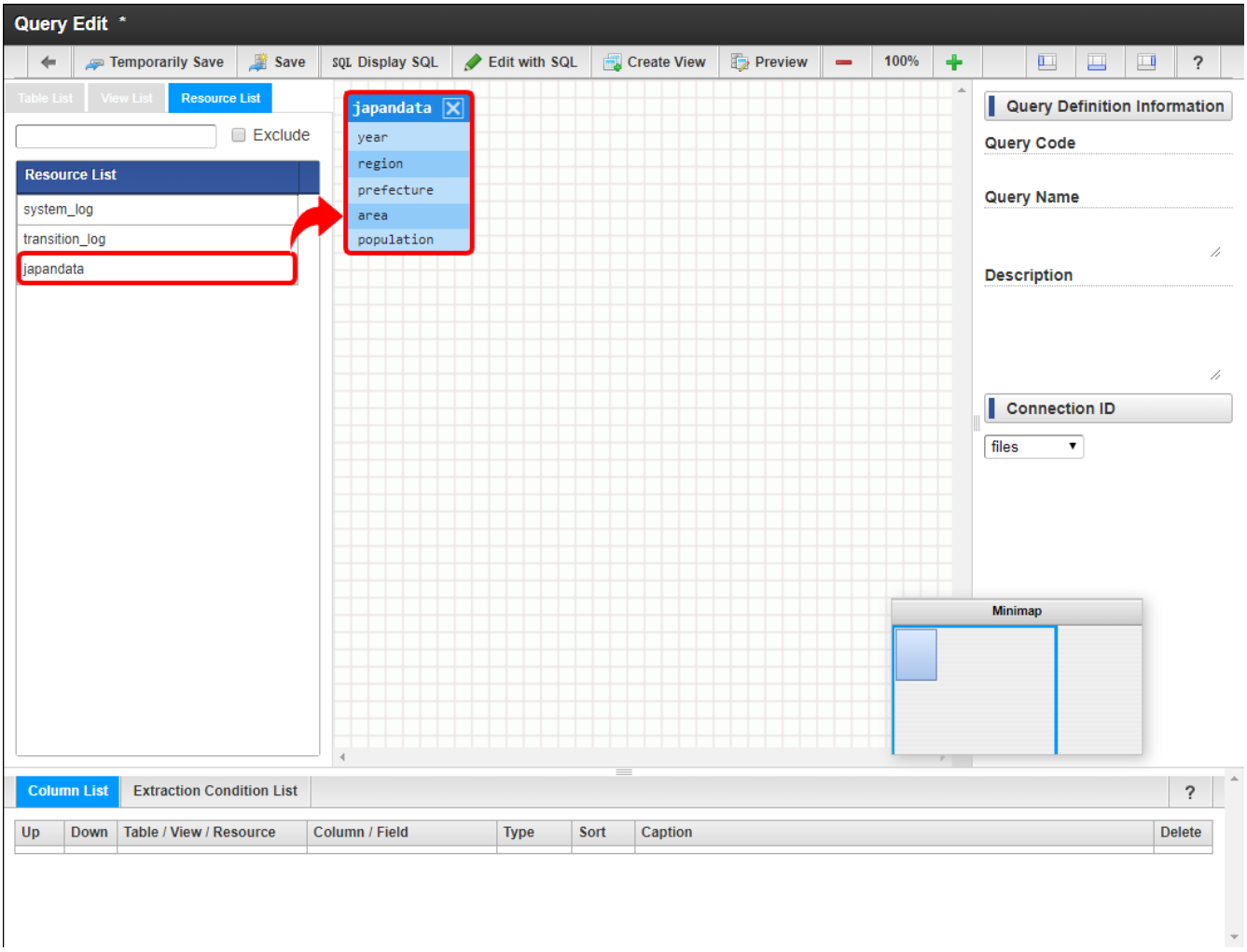


コラム

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。



カ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。

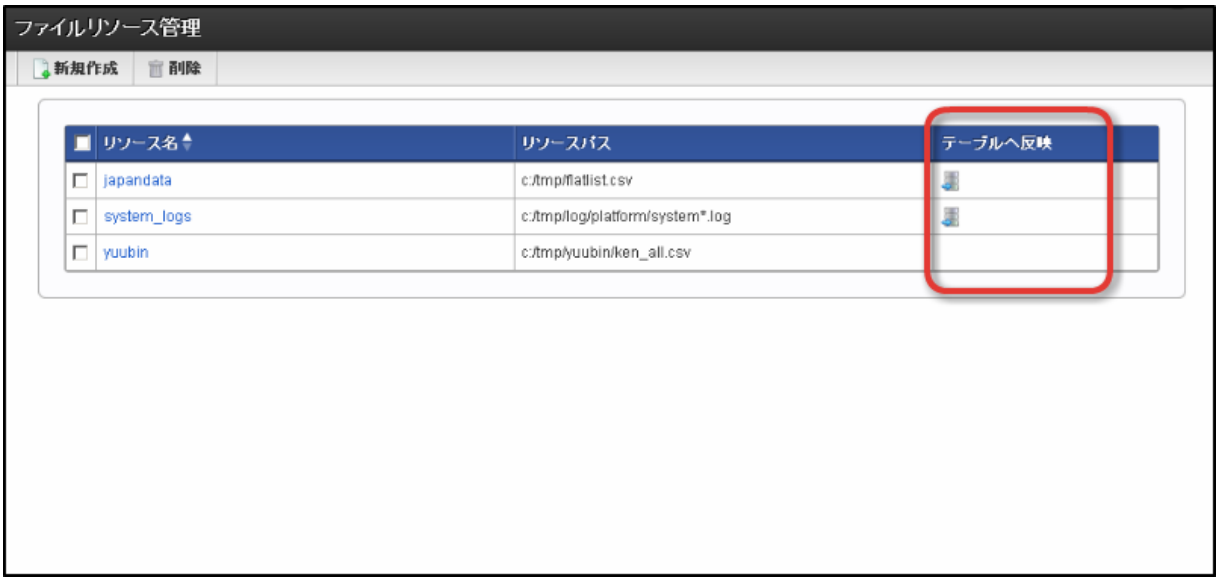
ファイルリソースとRDBの連携

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。

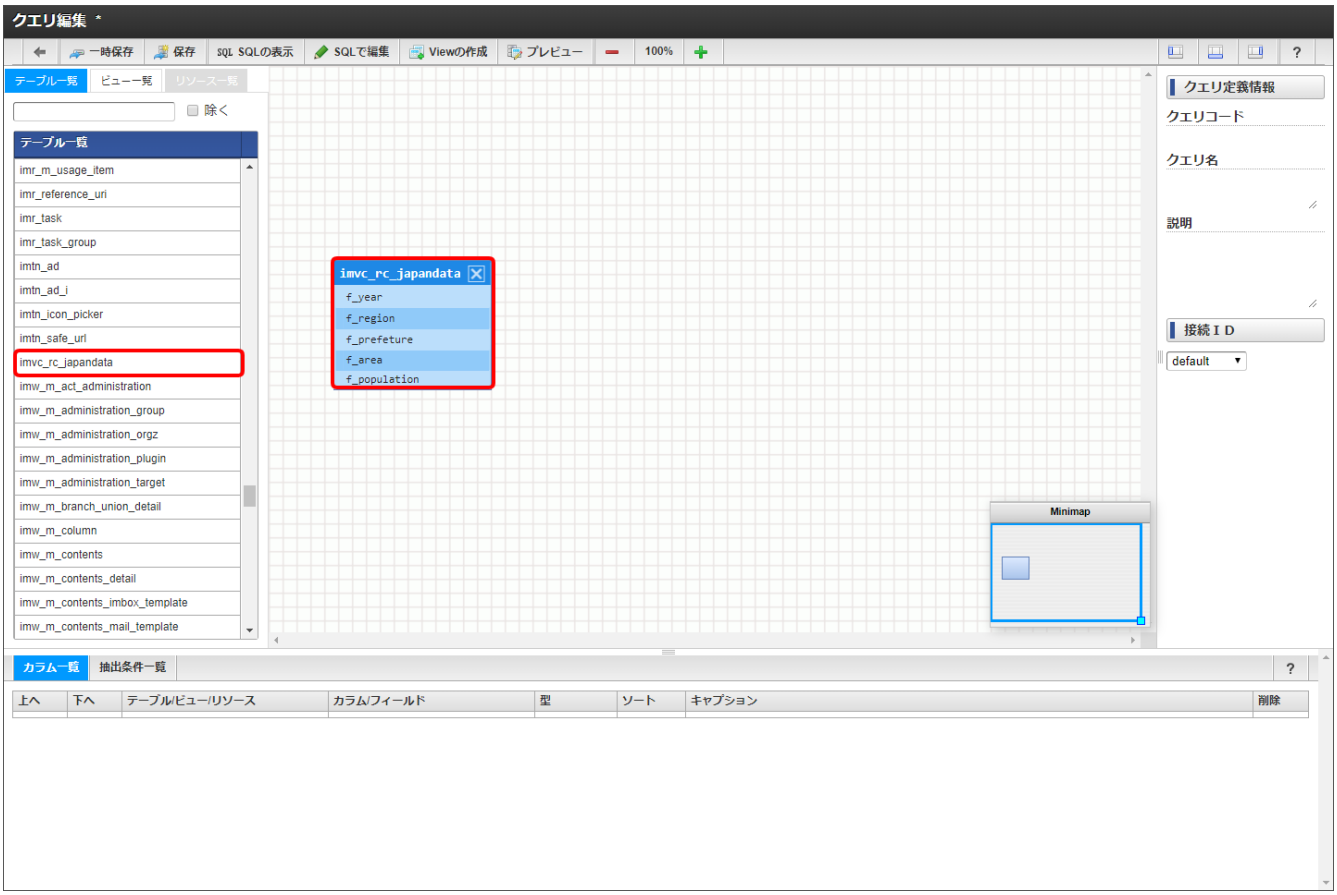
ファイルリソース登録の際に、「データベースとの連携」にチェックを入れて登録します。

> **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。



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。

注意

以下の条件を満たす場合、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 %コンテキストパス%/WEB-INF/conf/viewcreator-config.xml に新しい外部データソースの設定が必要です。

- 2017 Summer(Quadra) 以前の環境から 2017 Winter(Rebecca) 以降にアップデートした環境であること
- viewcreator-config.xml を IM-Juggling で出力していること

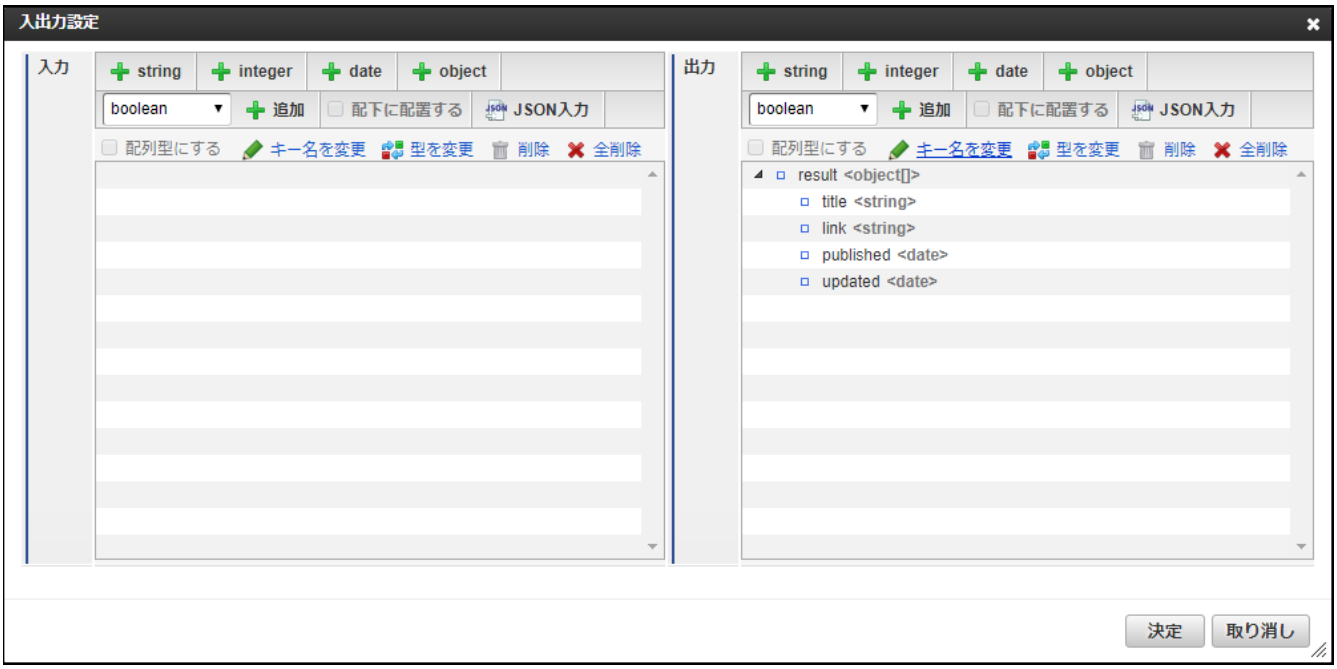
%コンテキストパス%/WEB-INF/conf/viewcreator-config.xml の<data-source-list>タグに以下の設定を追加します。

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

viewcreator-config.xml の詳細については「設定ファイルリファレンス」-「ViewCreatorの設定」を参照してください。

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。

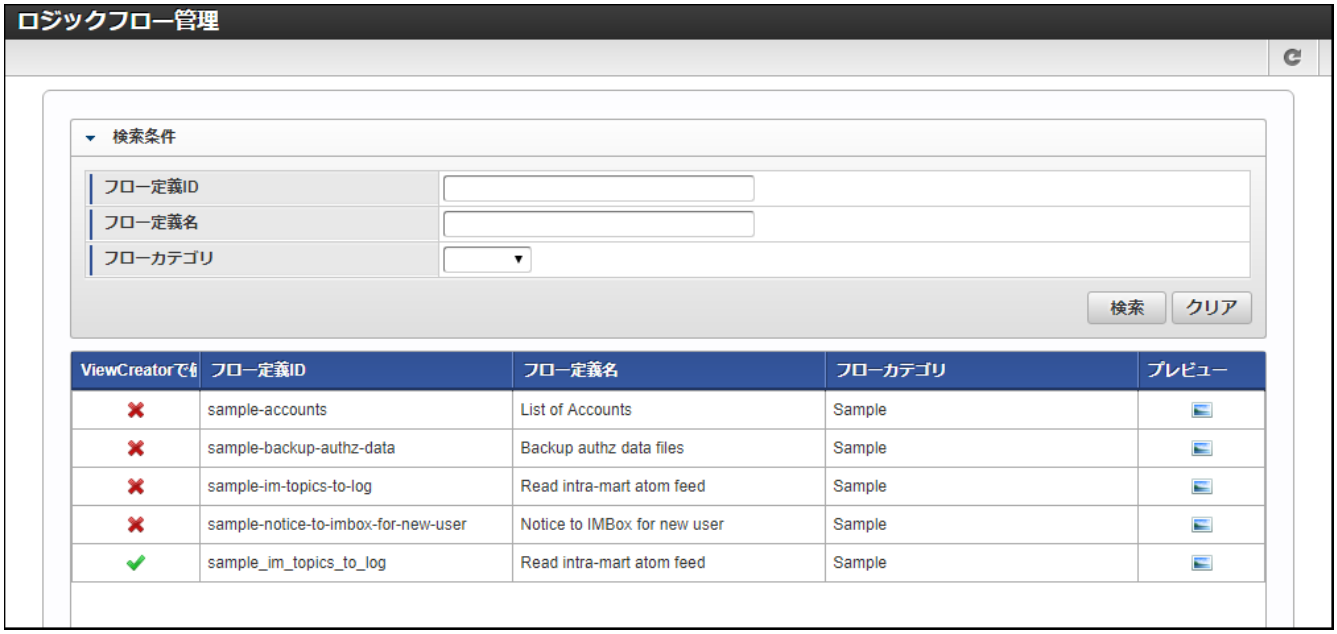
図：「ロジックフロー管理」画面 - 初期表示

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	（ ）アイコンをクリックし、（ ）にすることでリソースを登録します。 （ ）アイコンをクリックし、（ ）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。



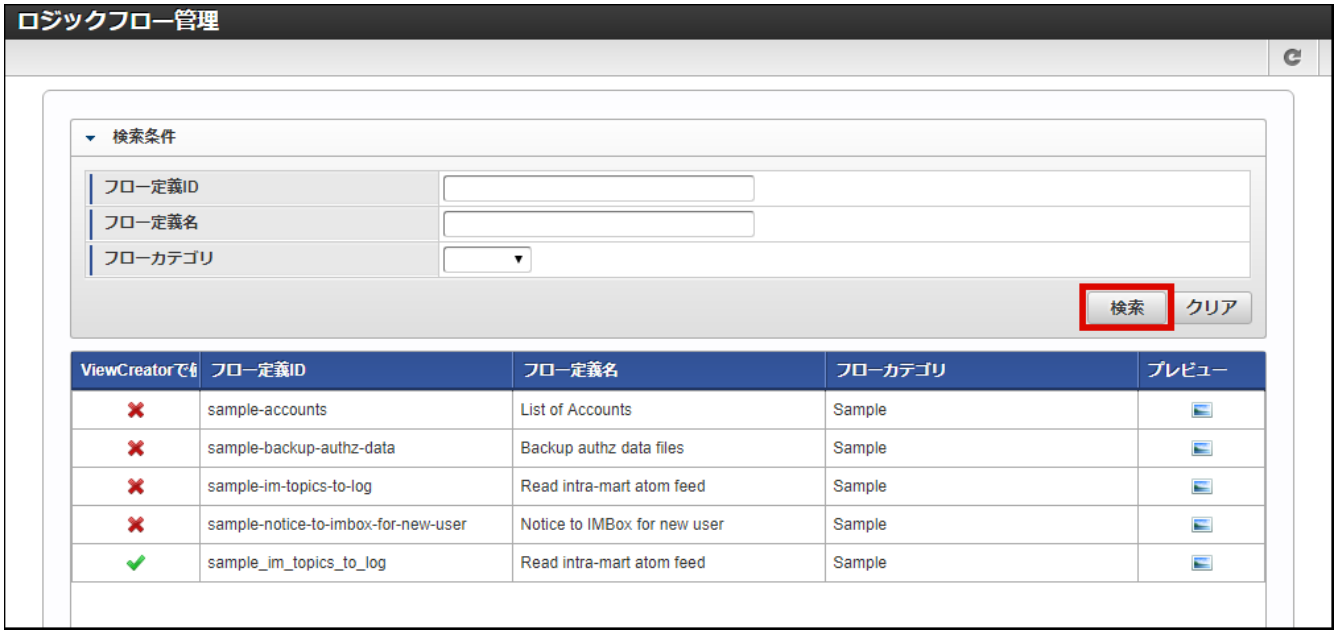
<画面項目>

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。

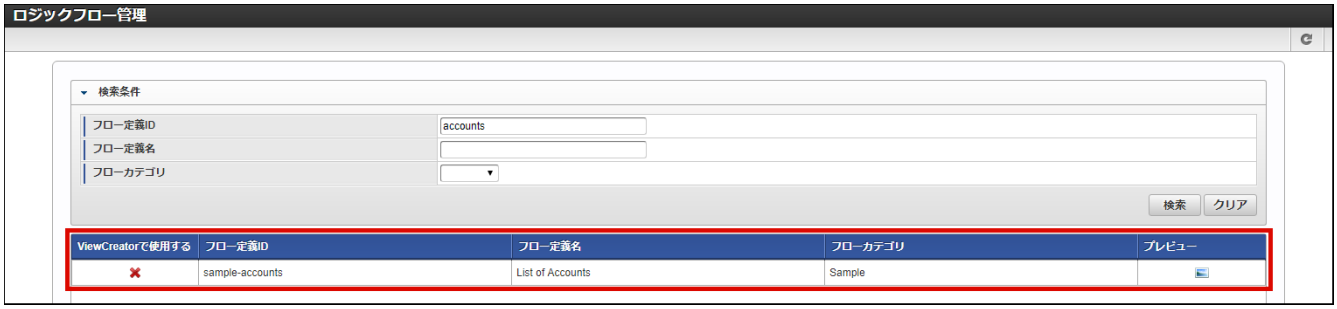
コラム

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。



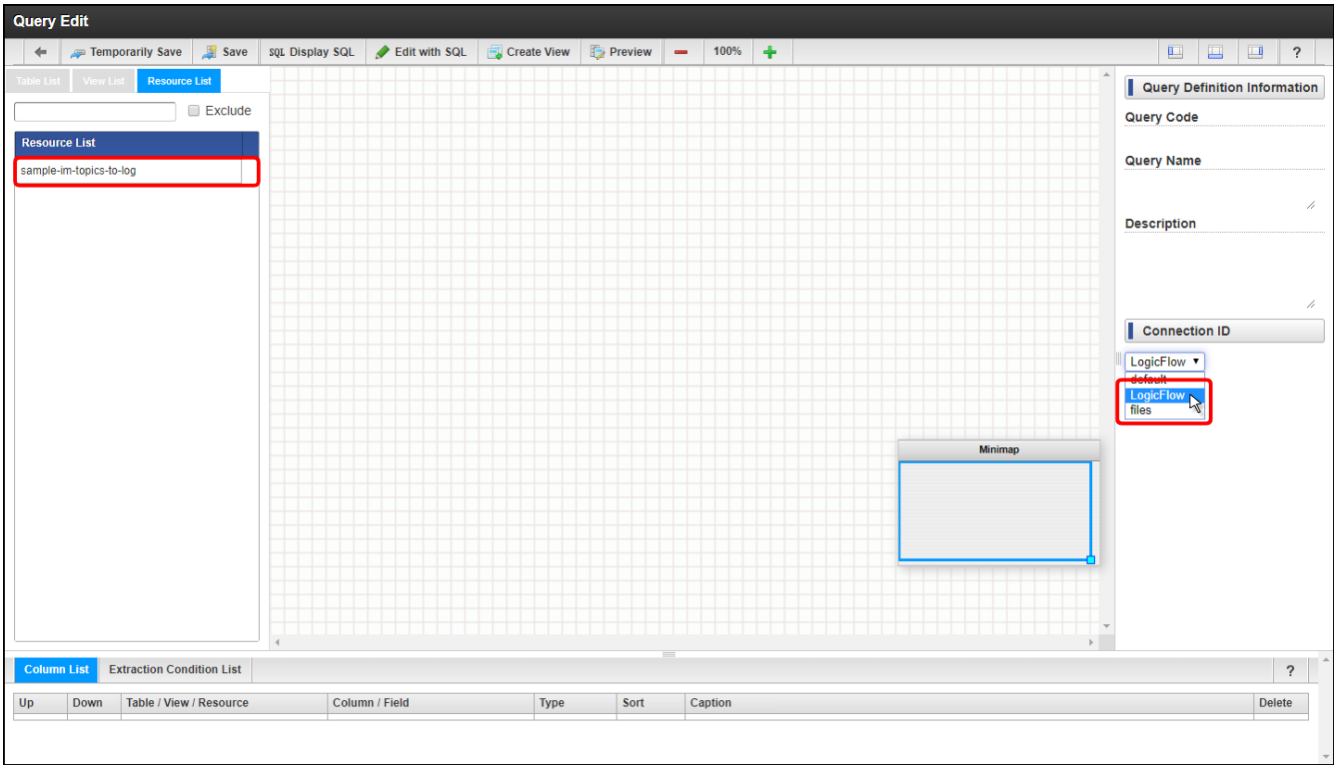
2. 検索結果が表示されます。



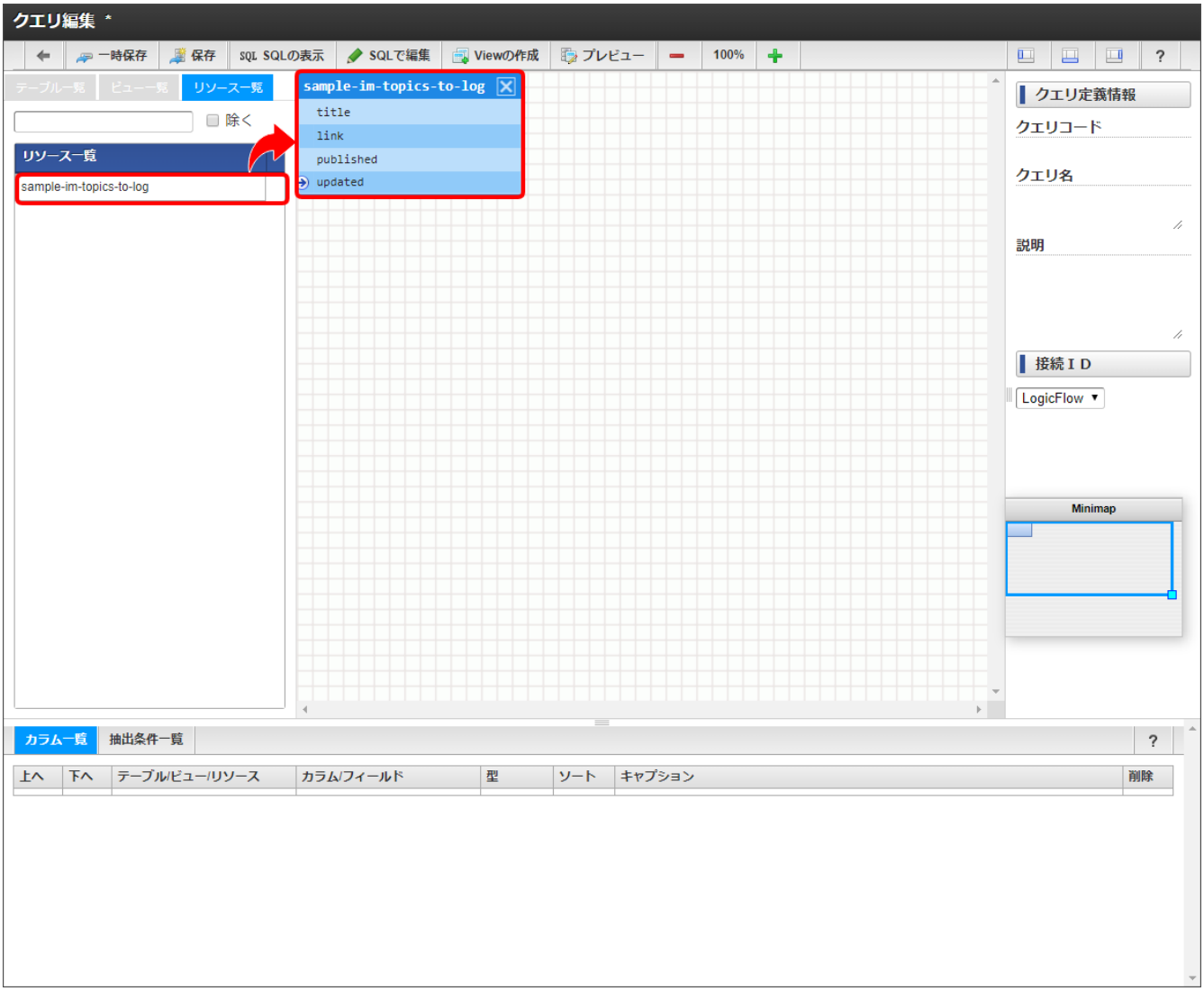
> **コラム**
>
> フローカテゴリ選択時の検索結果には、選択されたフローカテゴリの直下に存在するフローのみが表示されます。

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。

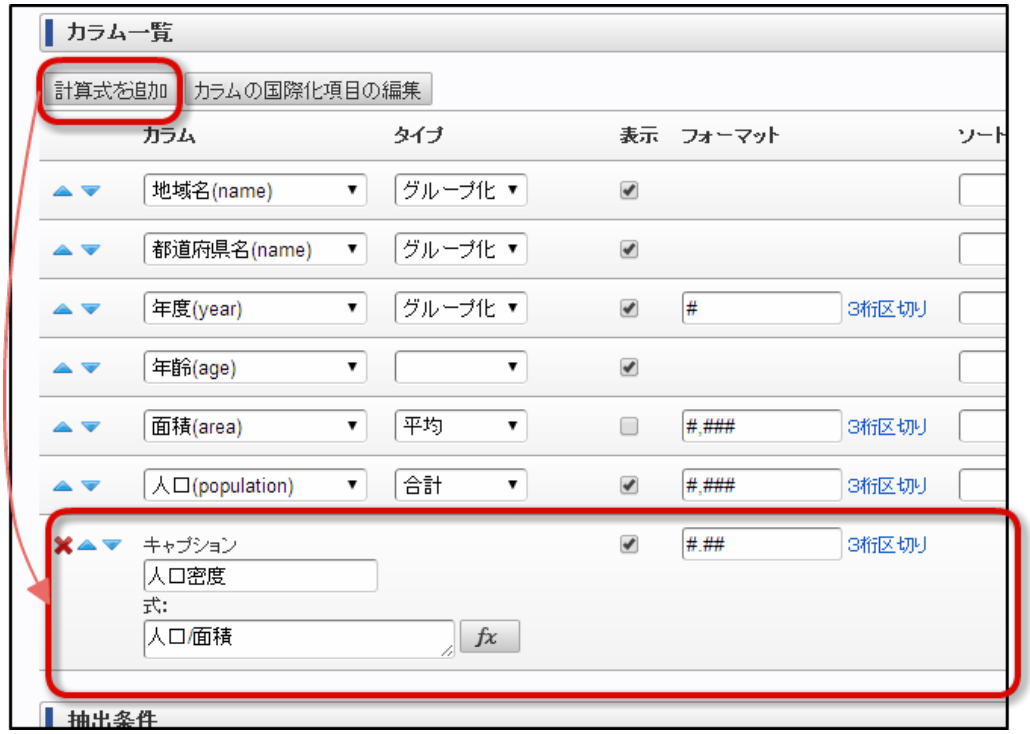
計算式カラムの設定

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。



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。

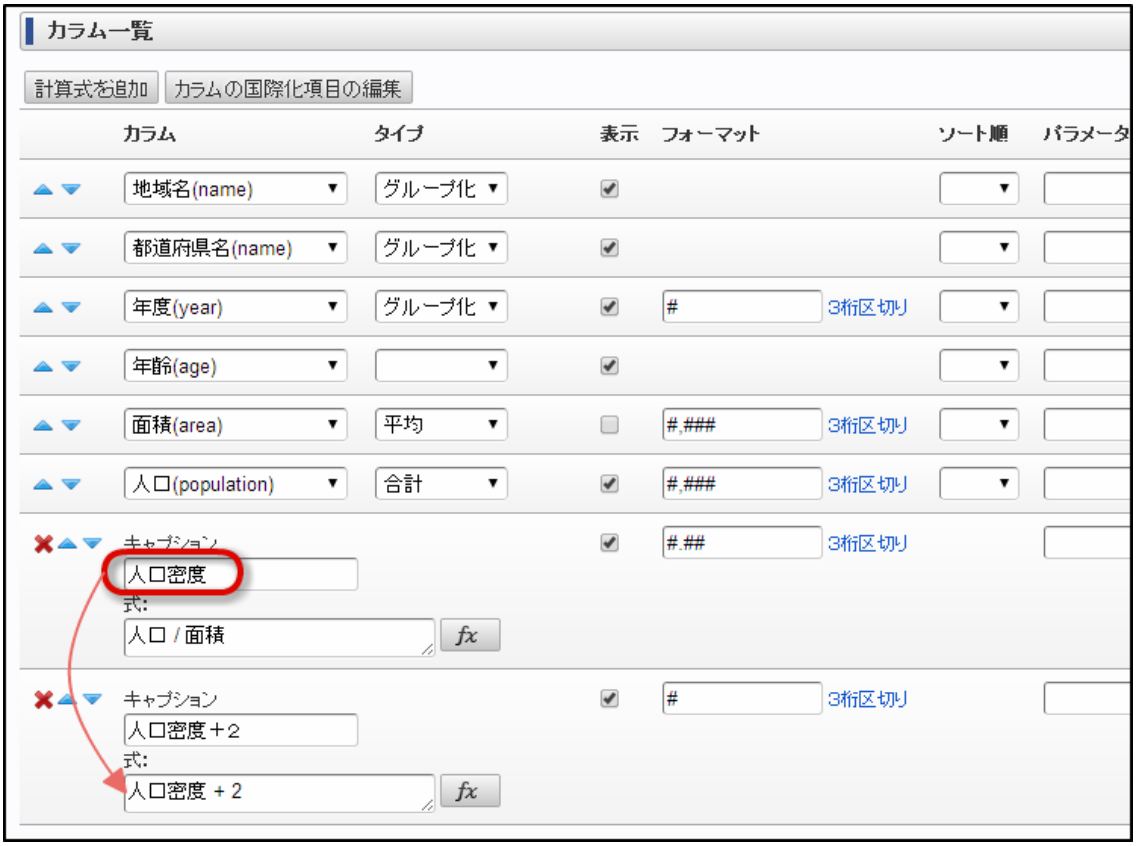
キャプション	カラムの列タイトルとして使用されます。
式：	計算式には他のカラムの「カラム名」を使用できます。 データ参照表示時には、該当するカラムのデータが展開されます。

実行結果

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。



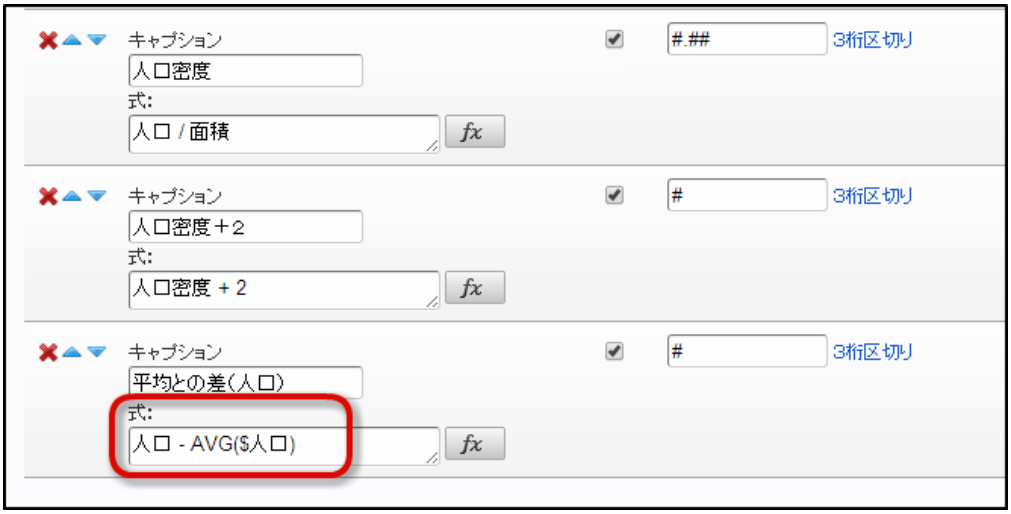
実行結果

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。

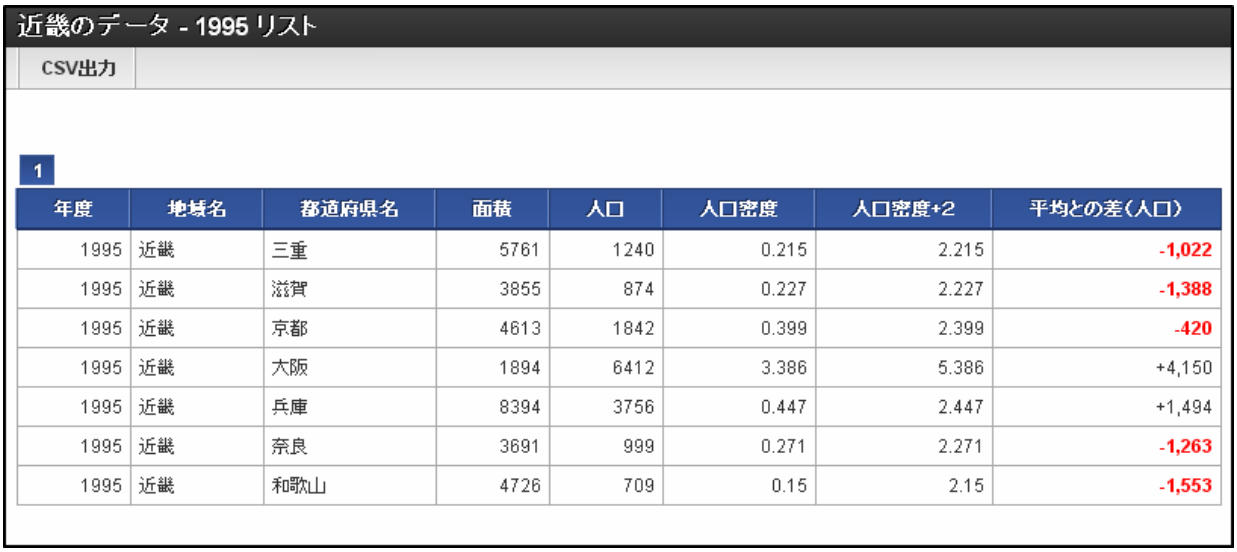
計算式中で使用可能な関数が用意されています。



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

クエリで定義された型	計算式で利用した場合の型
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	ー
ー	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。

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みます。

式入力エリアに関数が追加されます。

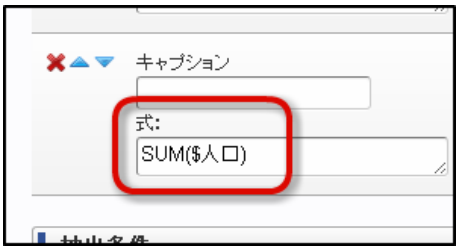
関数の引数の部分を有効なカラム名に書き換えます。

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。

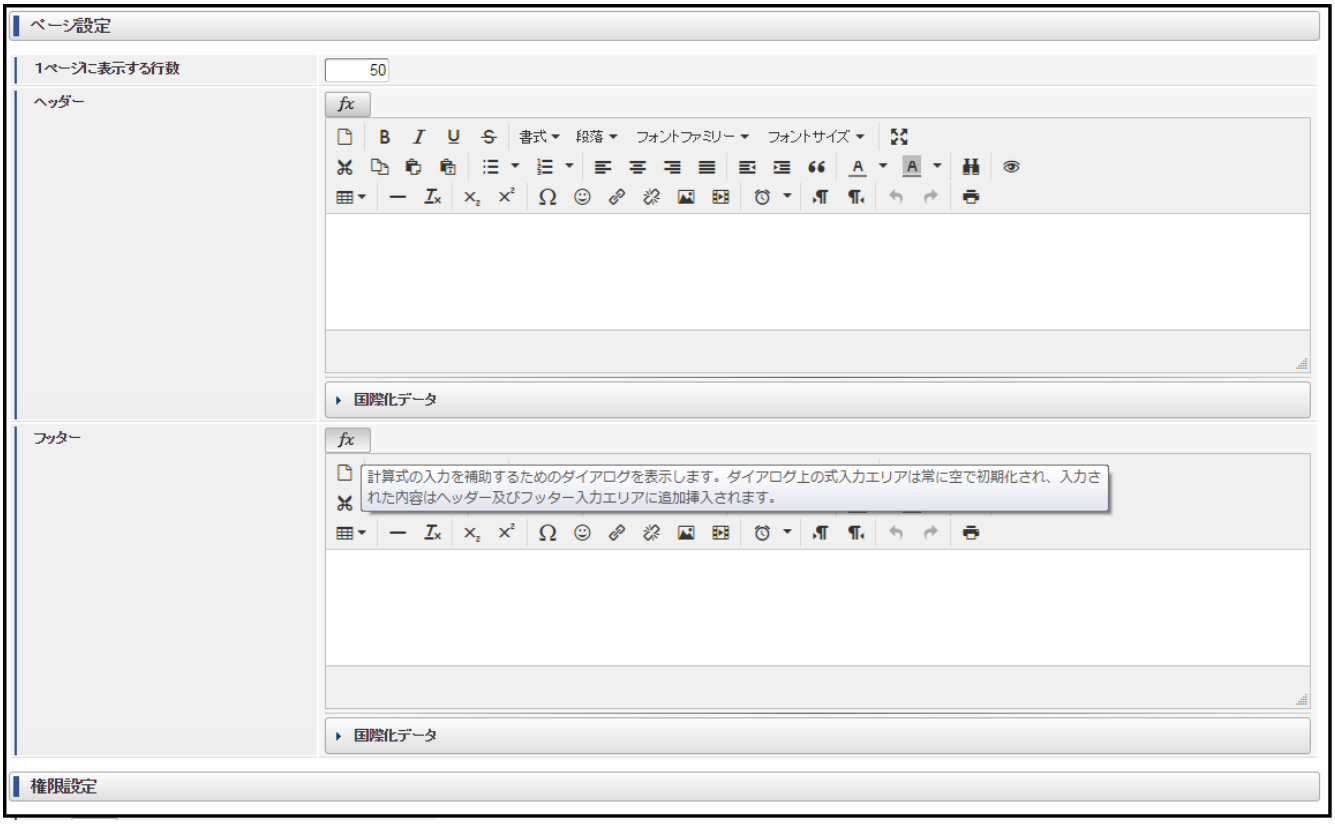
> **コラム**
>
> カラムの指定にはキャプションの代わりに内部的に生成されたコードを使用することもできます。
> キャプションを利用した場合、そのキャプションが変更されると式が実行できなくなってしまいますが
> コードを利用するとキャプションの変更に影響されることなく式の実行が可能です。
> 但し式の可読性が落ちるため、適宜使い分けてご利用ください。

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。



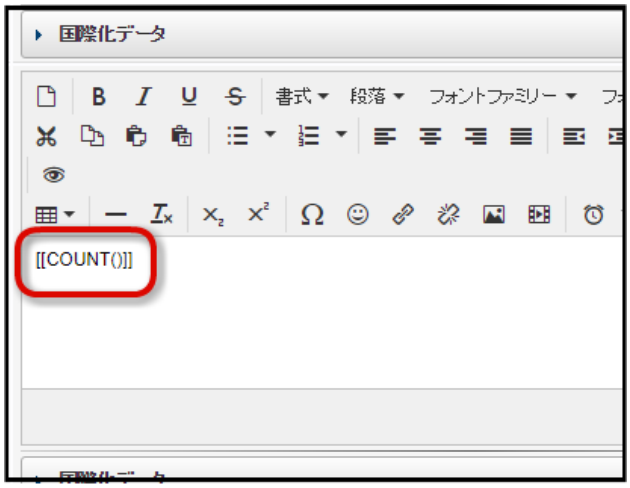
ヘッダとフッタの計算式設定

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）

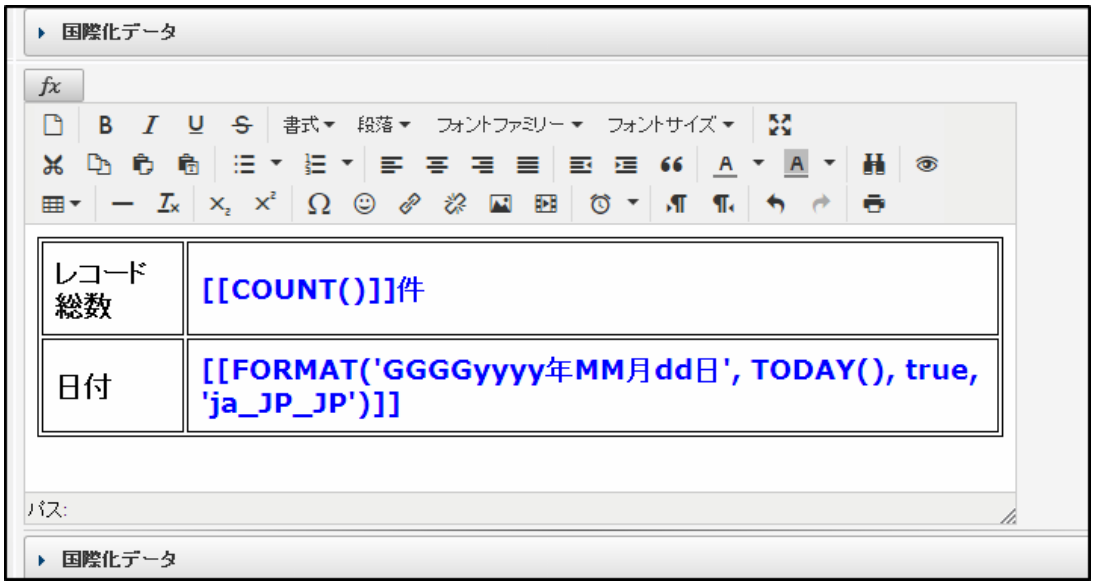
設定例	実行結果例
レコード総数：[[COUNT()]]件	レコード総数：110件

> **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。

リッチテキストと計算式

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



- 表示例

> **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...)

> **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

引数を要素に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。

SWITCHMULTI (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
> SWITCHMULTI関数は2022 Spring(Eustoma)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

判定対象値が文字列の場合
　カンマで連結された判定対象値を分割した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カンマで連結して返します。
　例： SWITCHMULTI (都道府県コード, “1”, “東京”, “2”, “神奈川”, “3”, “埼玉”, “その他”)
　都道府県コード が 1,3,4 である場合は、東京,埼玉,その他 が返却されます。
判定対象値が文字列以外の場合
　SWITCH関数が実行され結果が返却されます。

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
第二引数には、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>useDoubleQuotation4AllStringValues</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/useDoubleQuotation4AllStringValues	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。
useDoubleQuotation4AllStringValues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
2. 設定ファイルで指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出力ジョブの登録

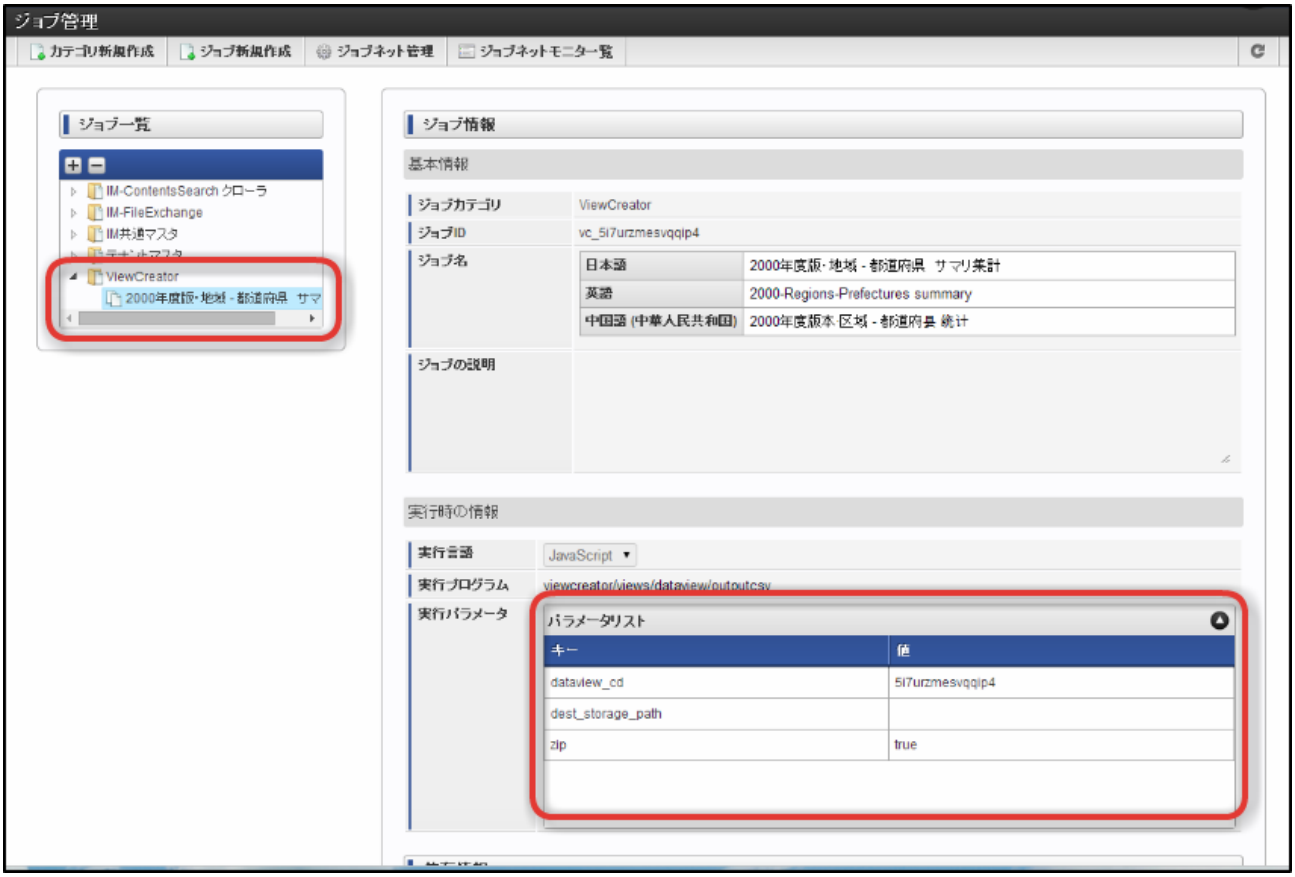
リスト集計および、サマリ集計のデータ参照からCSVファイルを出力するジョブを登録します。

> コラム
> CSV出力ジョブの登録は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出力ジョブの登録」を選択します。

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。

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。

  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。

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	543.98

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>
  <use-browser-print>true</use-browser-print>
  <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。

コラム

use-browser-printの設定は2023 Autumn(Hollyhock)から利用可能です。

use-browser-print	帳票出力にブラウザ印刷を利用する場合にtrueを設定します。 report-creator-class-nameやreport-create-commandを利用する場合はfalseを設定します。

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。 この設定はreport-creator-class-nameやreport-create-commandを利用して帳票出力する場合のみ有効です。

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>
```

> **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。

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>
```

グラフ集計で「凡例」の表示方法を変更する

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。

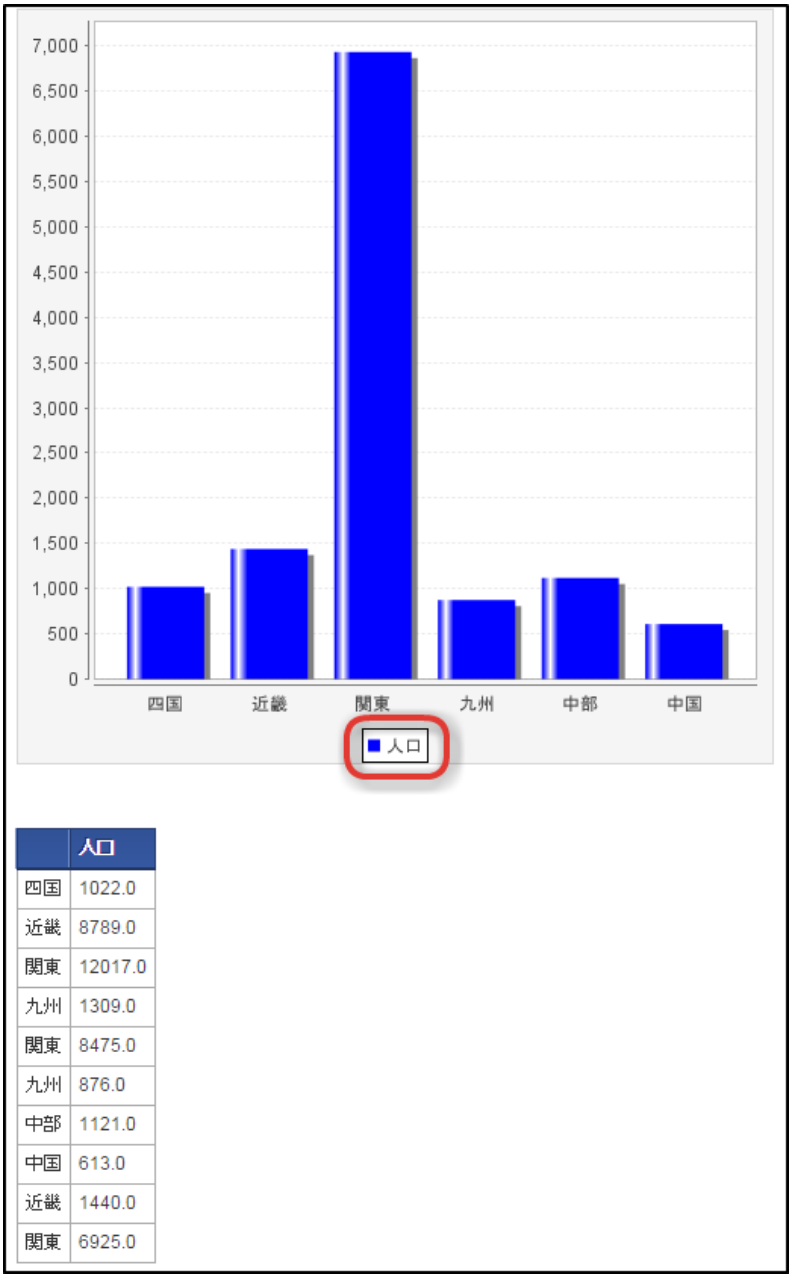
コラム

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



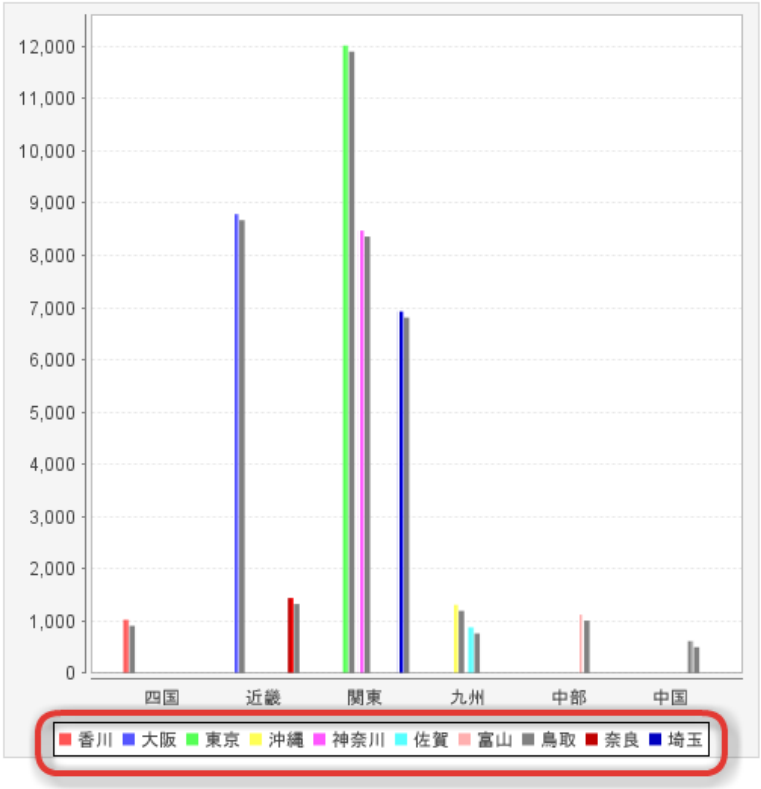
	人口
四国	1022.0
近畿	8789.0
関東	12017.0
九州	1309.0
関東	8475.0
九州	876.0
中部	1121.0
中国	613.0
近畿	1440.0
関東	6925.0

凡例として「人口」というデータが表示されています。

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

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。



	香川	大阪	東京	沖縄	神奈川	佐賀	富山	鳥取	奈良	埼玉
四国	1022.0									
近畿		8789.0							1440.0	
関東			12017.0		8475.0					6925.0
九州				1309.0		876.0				
中部							1121.0			
中国								613.0		

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。
画面上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

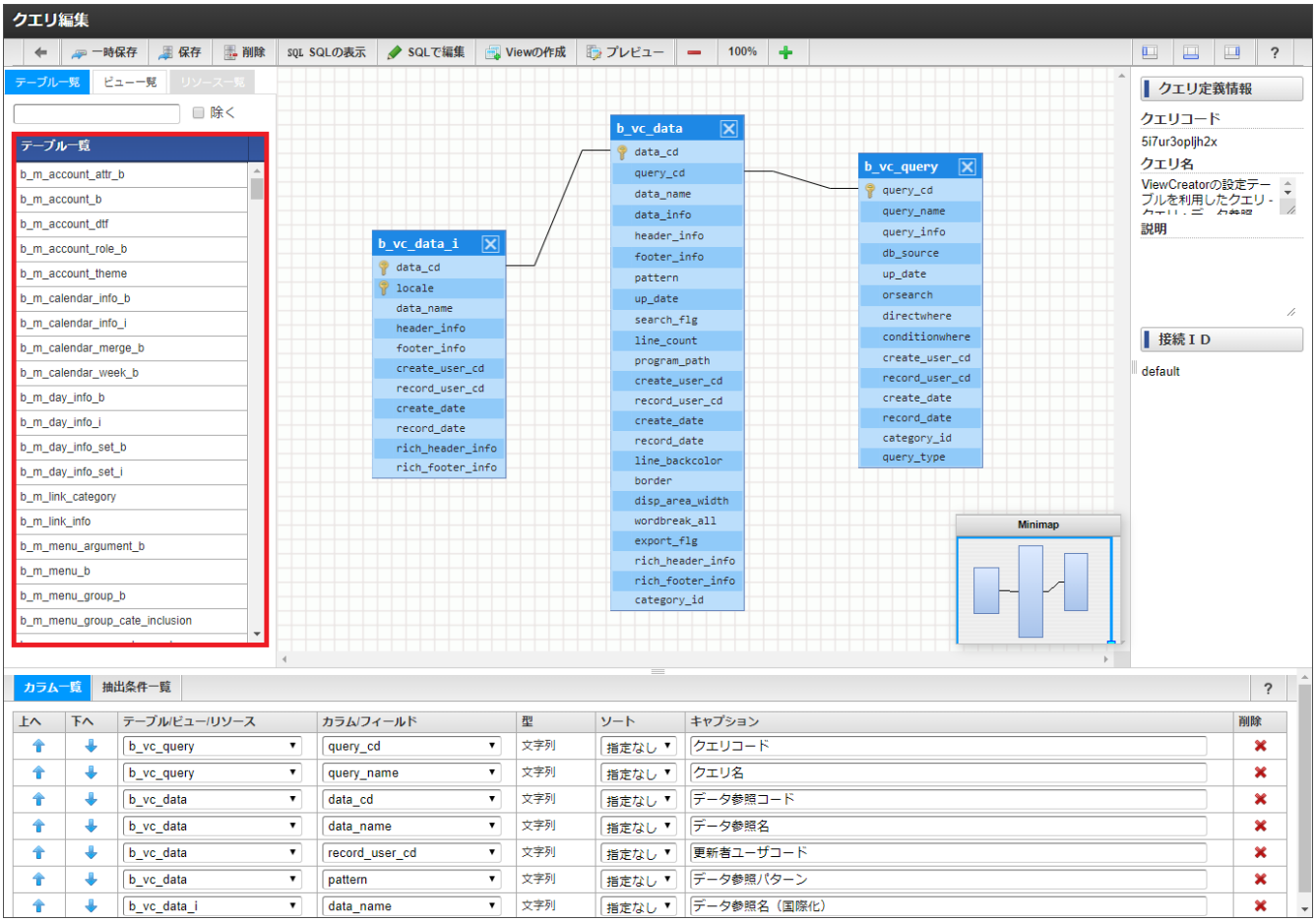
クエリ編集画面のテーブル一覧に表示されるテーブルにはアクセス権（参照権）を設定できます。

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。



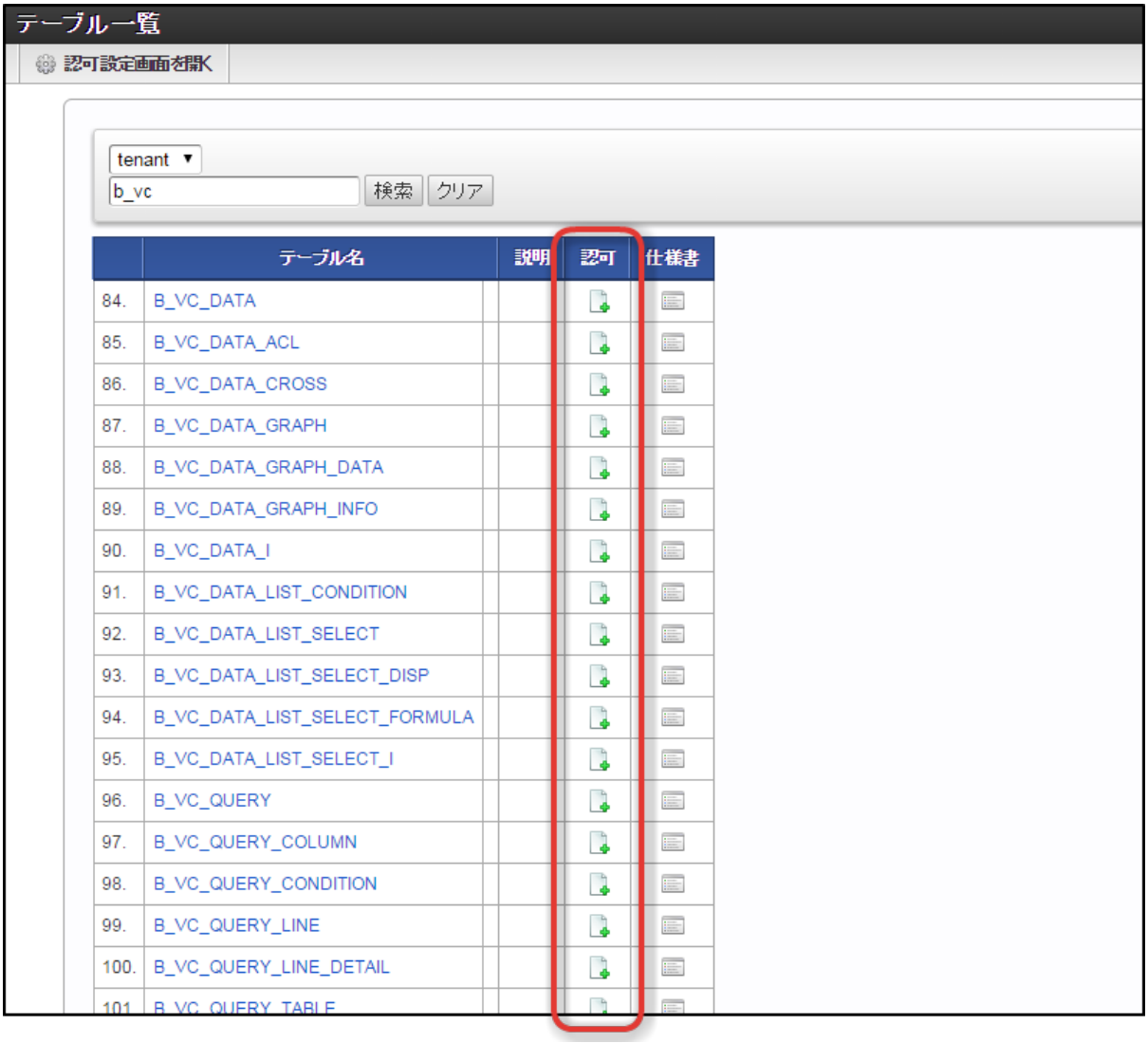
目次

- 設定の手順

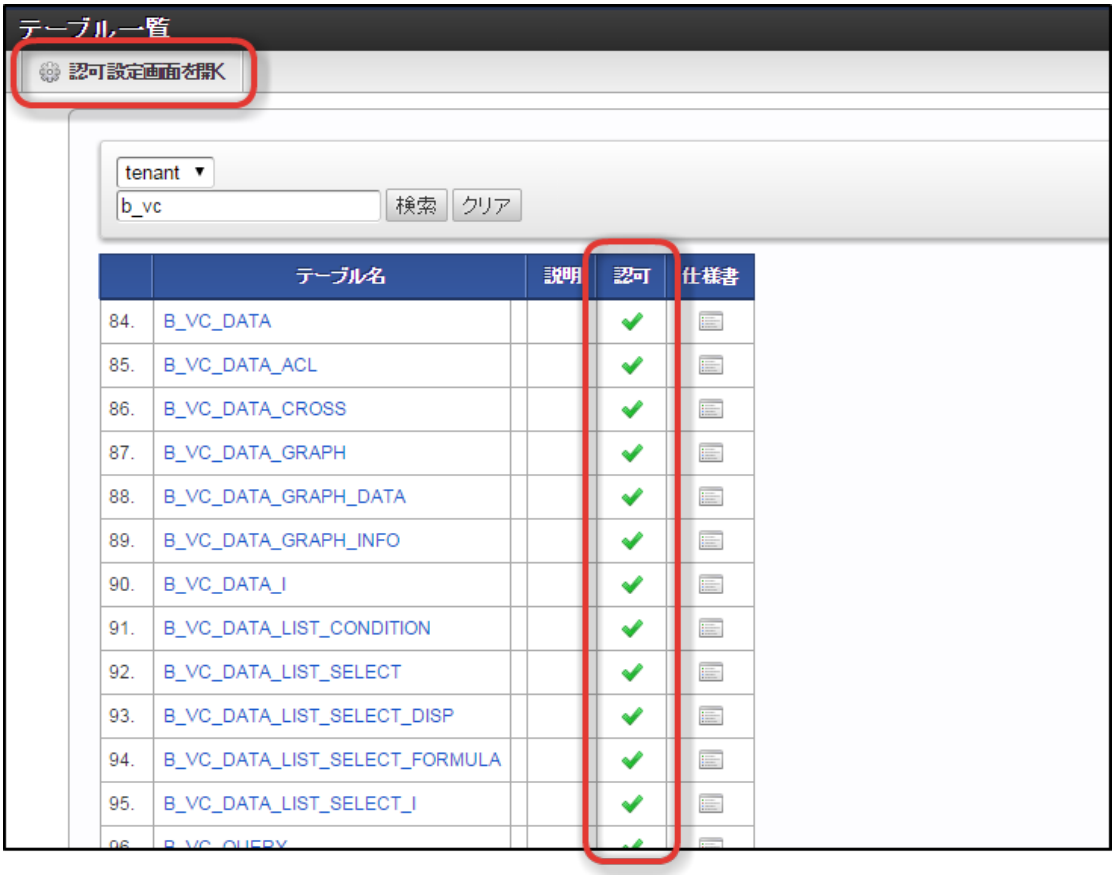
設定の手順

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。

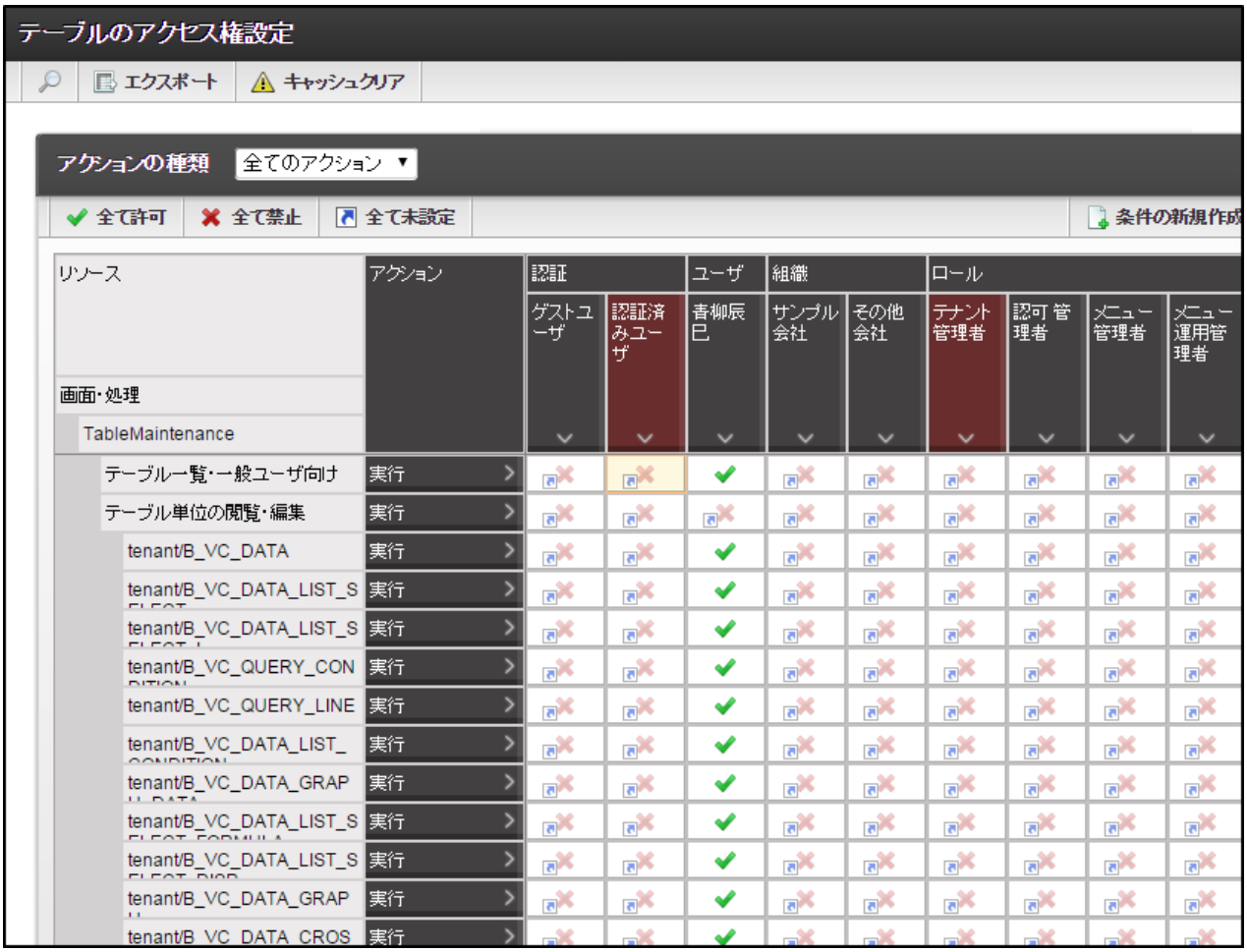


「認可」列のアイコンをクリックして認可リソースの登録を行います。
登録が完了すると、アイコンが緑のチェックマーク表示に変わります。



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。

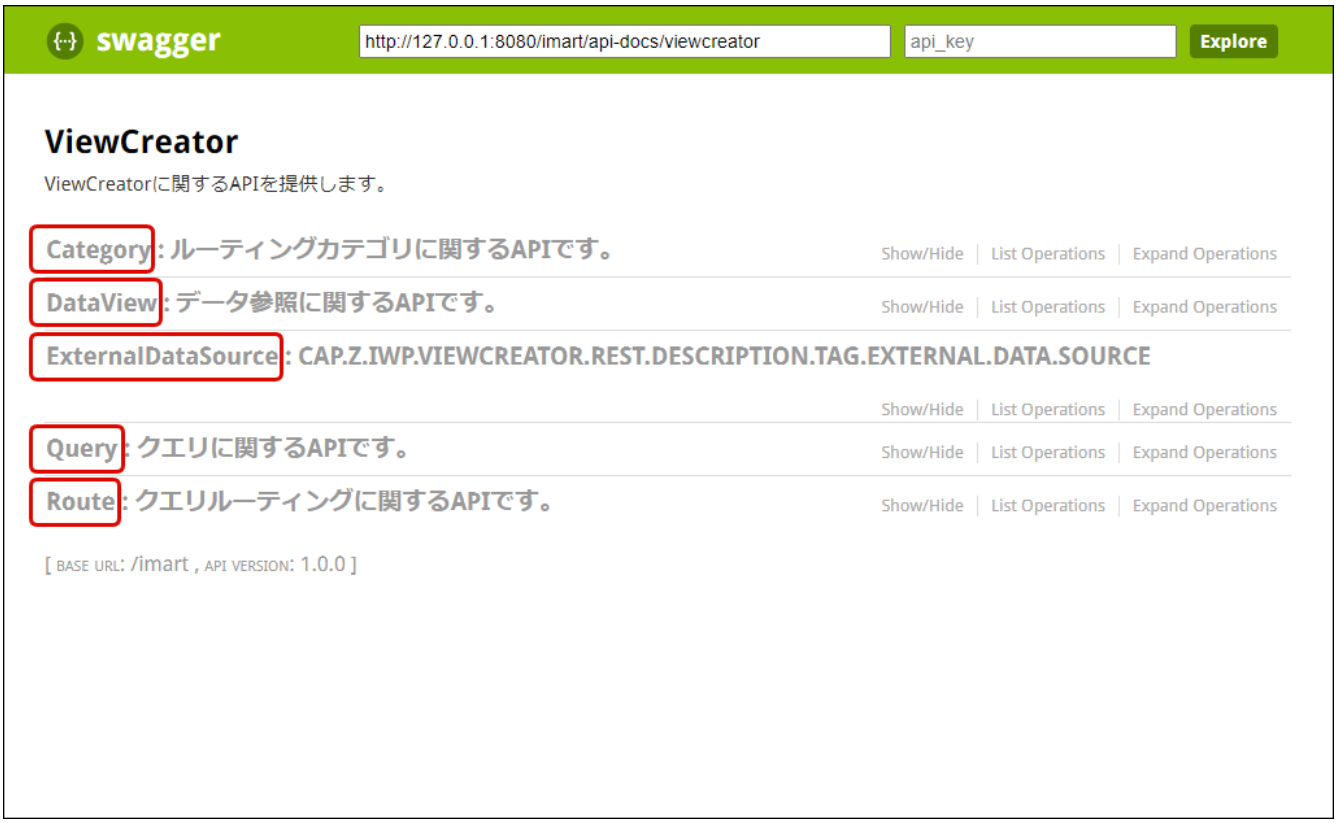
REST APIを参照する

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。

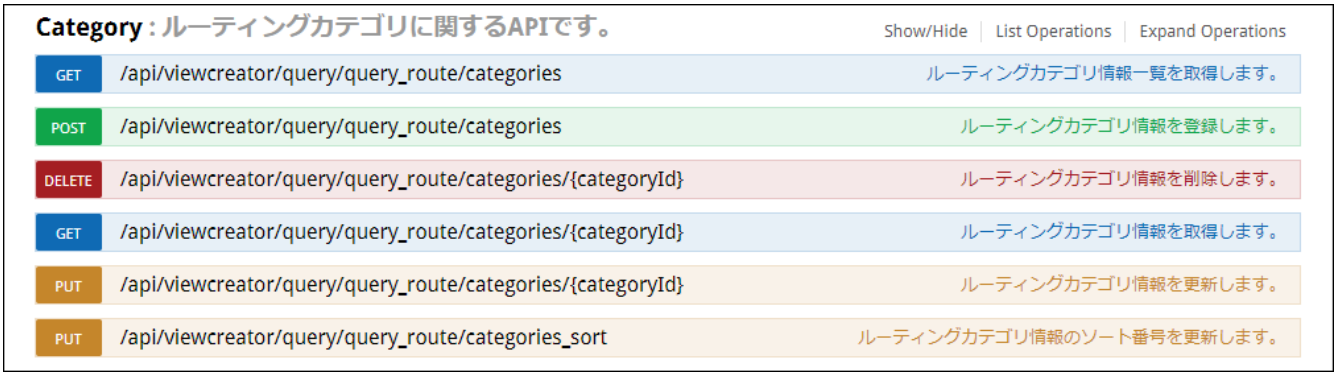
Swagger UIは、以下のURLから確認できます。

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.DESCRIPTION.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": [
      {
```

Response Content Type application/json

Parameters

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

Try it out!

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

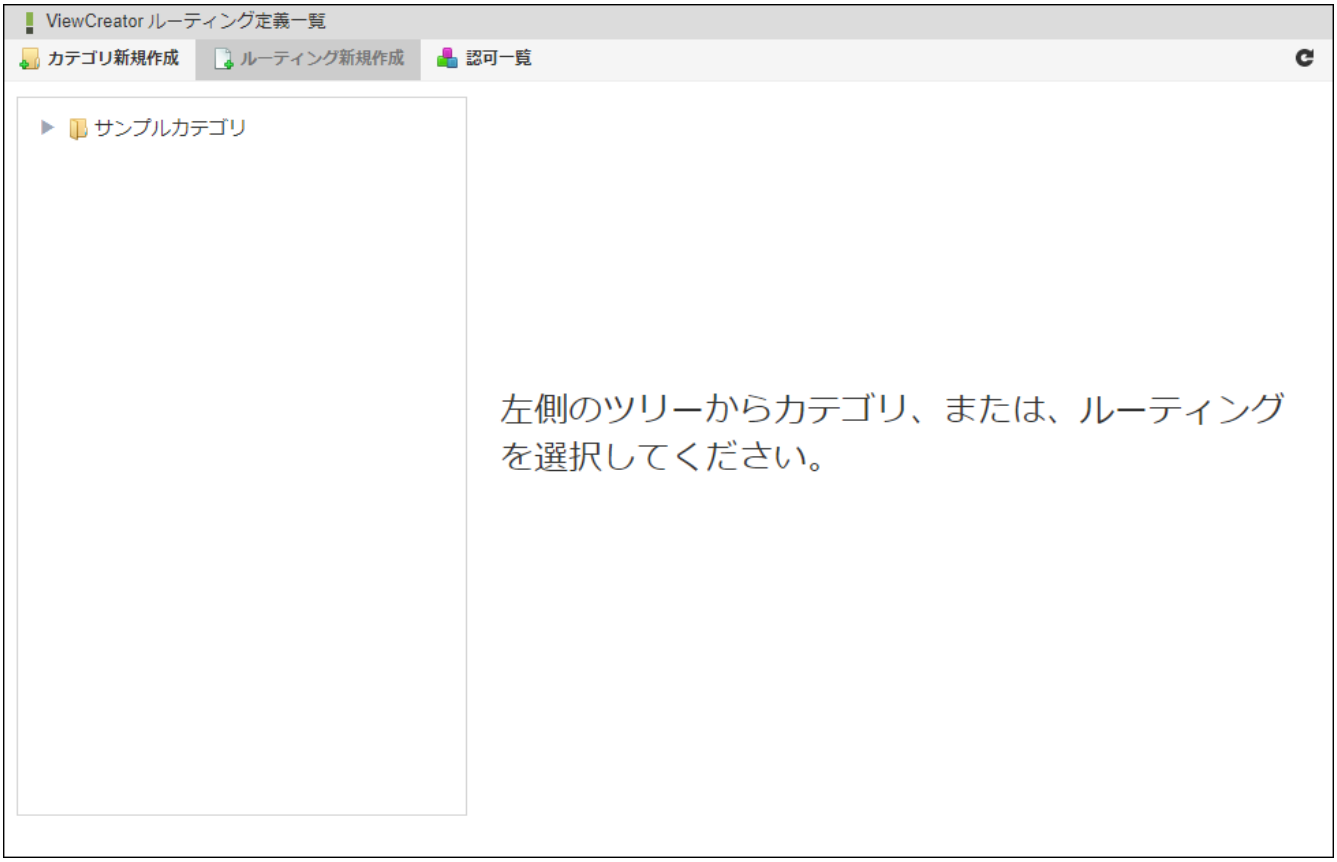
ルーティング定義作成機能は 2020 Winter(Azalea) から利用可能です。

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 OAuth スコープ設定
  - 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

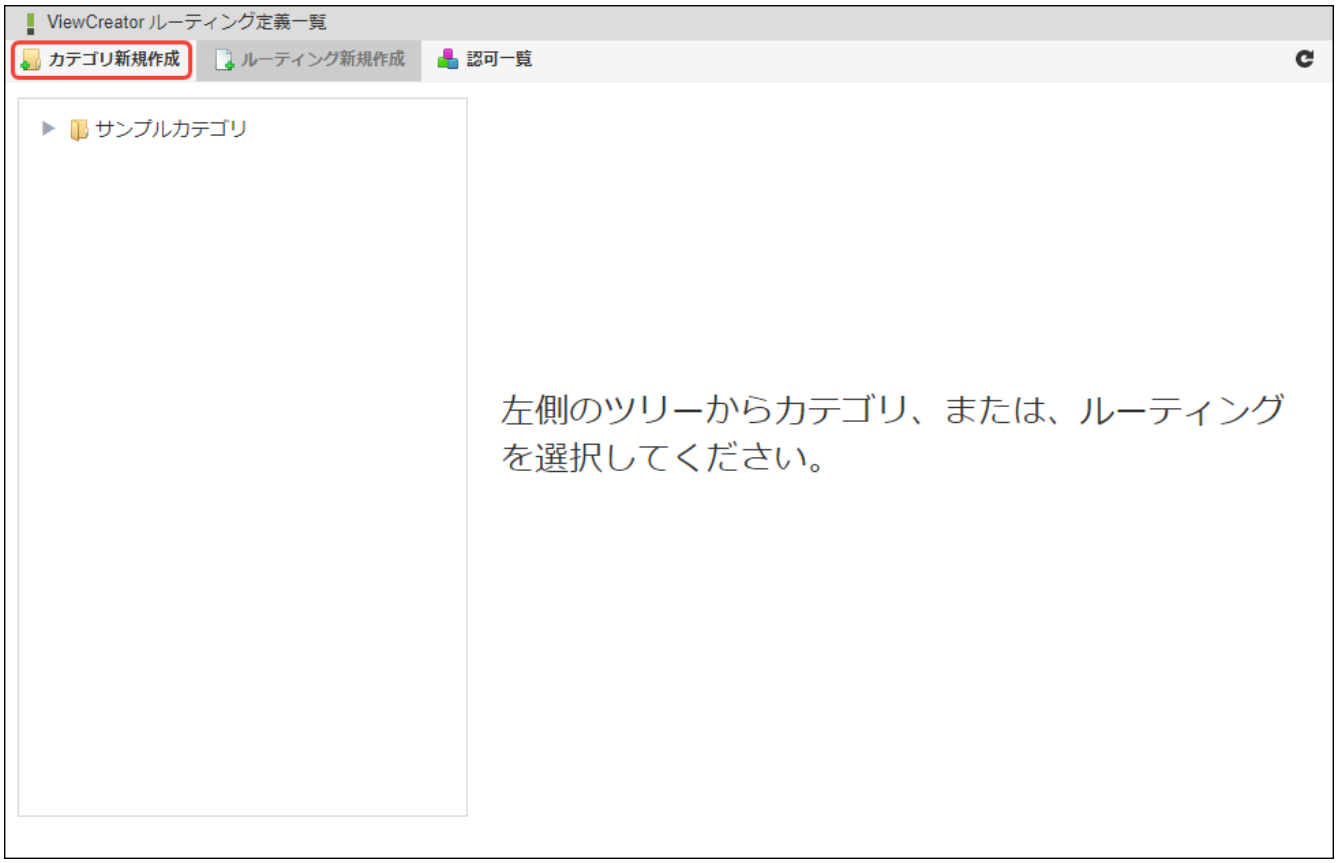
項目名	説明

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「*ルーティングカテゴリの作成*」を参照してください。
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「*ルーティングの作成*」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

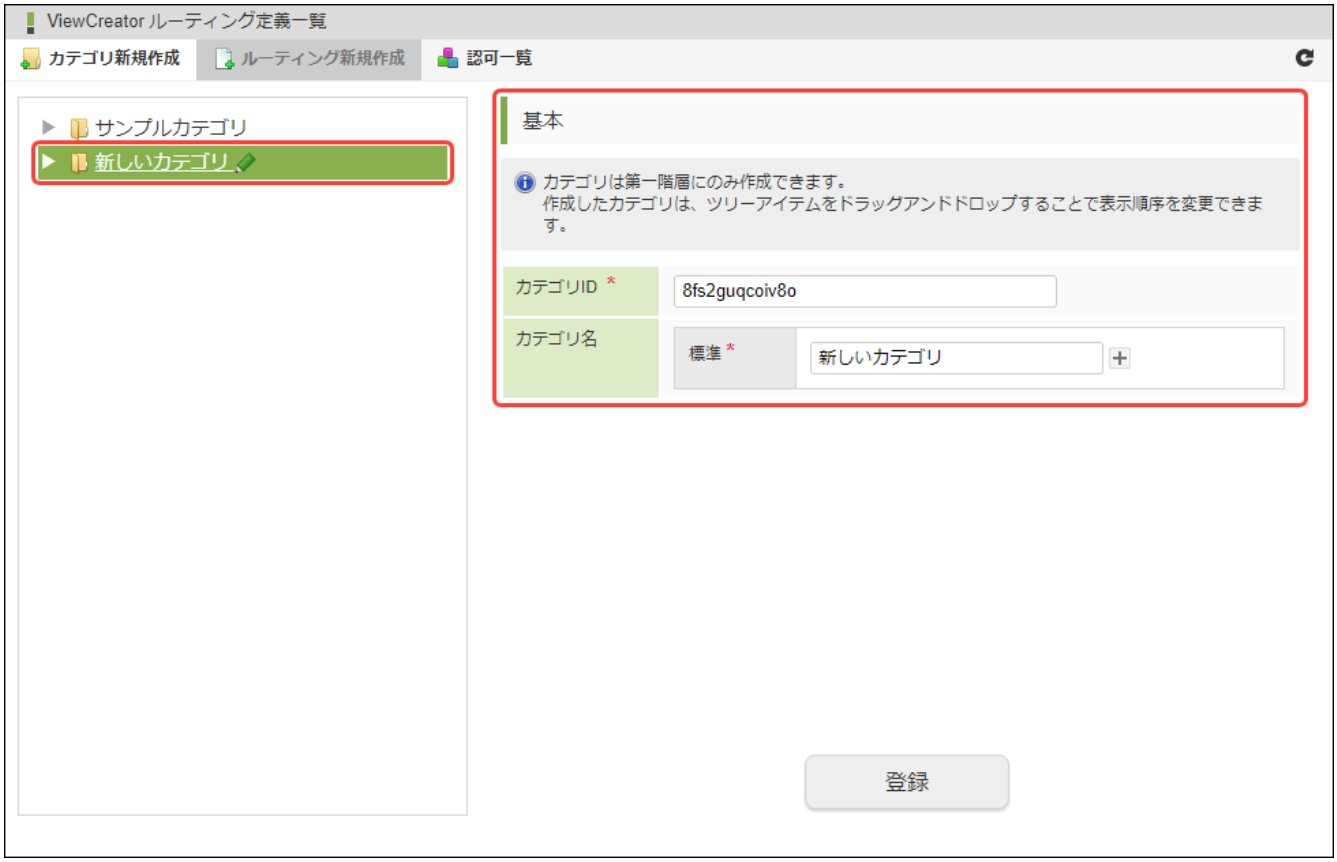
ルーティングはルーティングカテゴリに紐づけられる形で管理されているため、はじめにルーティングカテゴリを作成します。

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。



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。

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。

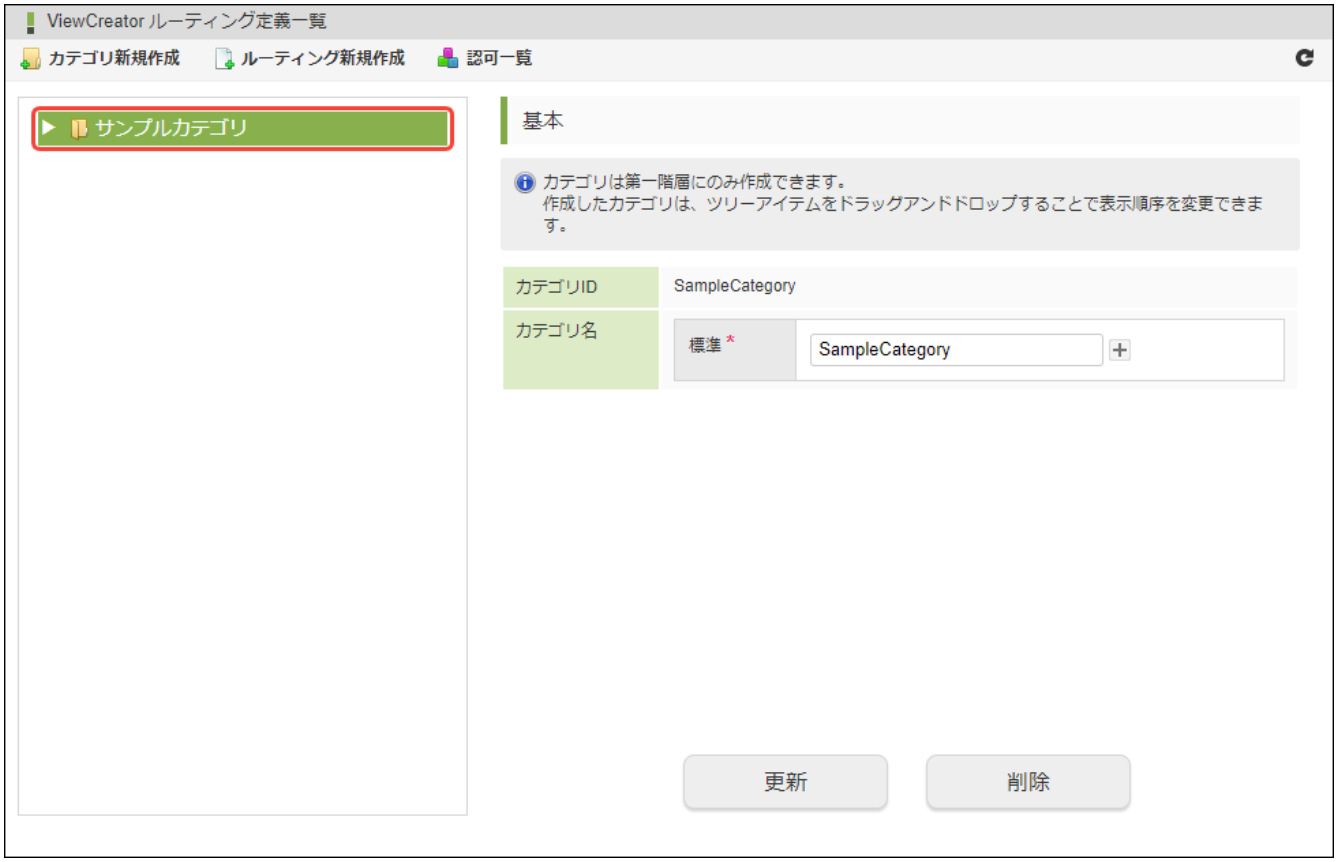
- 日本語（ja）ロケールが設定されているユーザの場合、「日本語」に設定したものが表示されます。
- 日本語（ja）ロケール**以外**が設定されているユーザの場合、「標準」に設定したものが表示されます。

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

ルーティングの作成

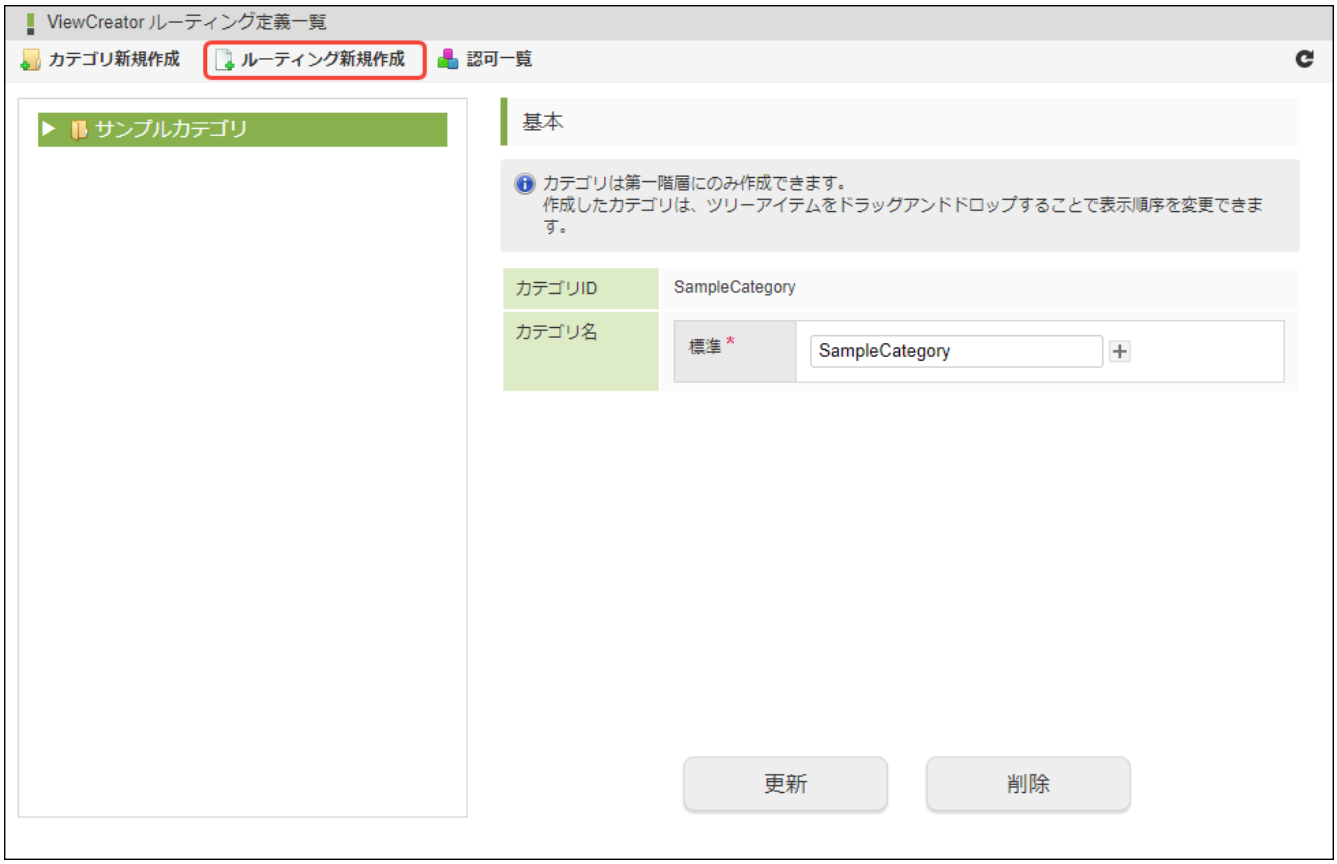
次に、ユーザがクエリを実行するための定義情報である「ルーティング」を作成します。

1. 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。



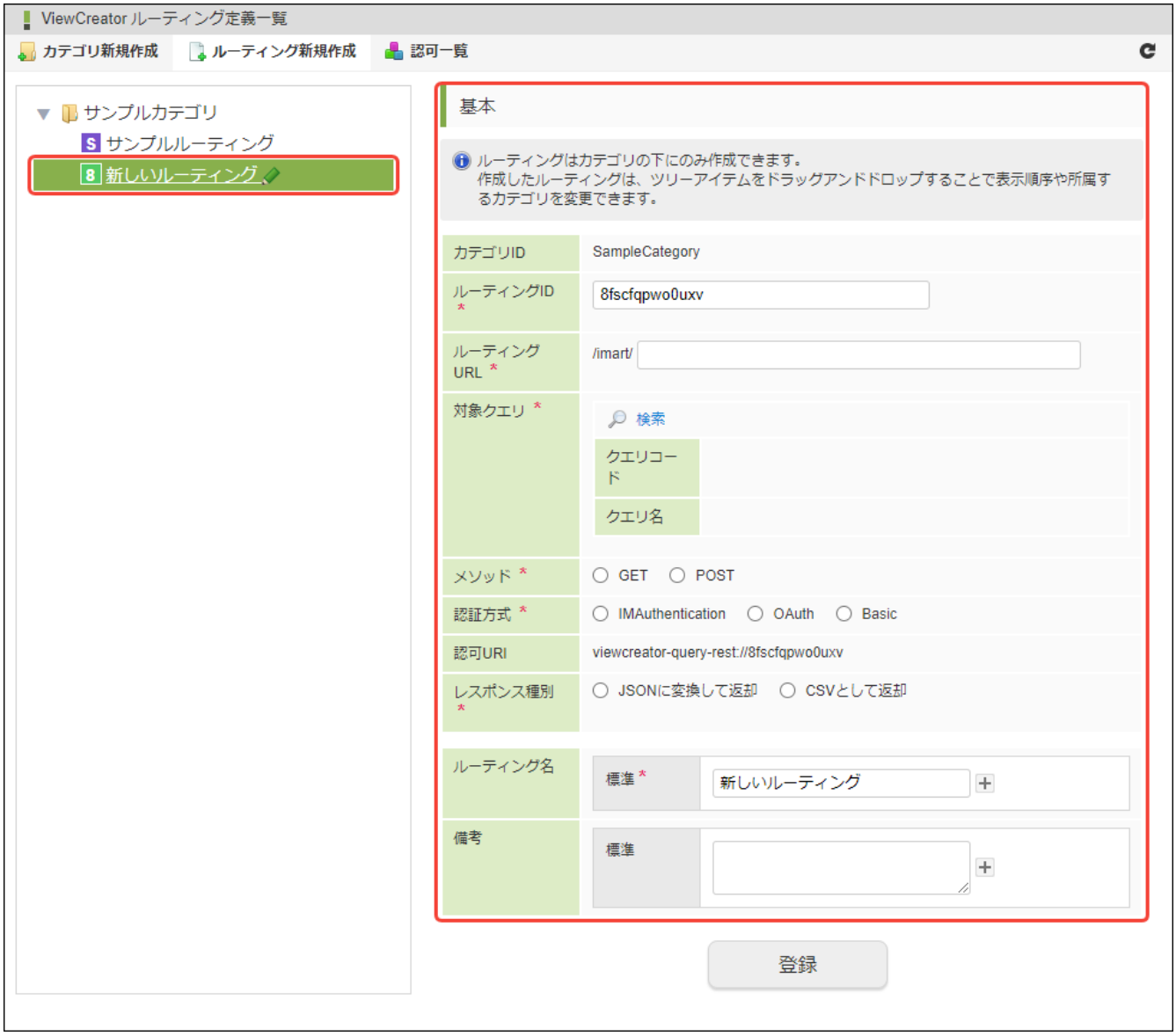
図：「ルーティング定義一覧」画面 - ルーティングツリー

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。



図：ルーティング情報

項目名	必須	説明
カテゴリID	ー	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「*ルーティングのクエリ設定*」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「*ルーティングの認証方式*」を参照してください。
OAuth スコープ	ー	OAuth認証を使用する場合のスコープを設定します。 「検索」をクリックして使用するOAuth スコープを選択してください。 認証方式で *OAuth* を選択した場合のみ設定できます。 詳細は「*ルーティングの OAuth スコープ設定*」を参照してください。

項目名	必須	説明
認可URI	ー	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「*ルーティングの認可設定*」を参照してください。
レスポンス種別	ー	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「*レスポンスボディ*」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	ー	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

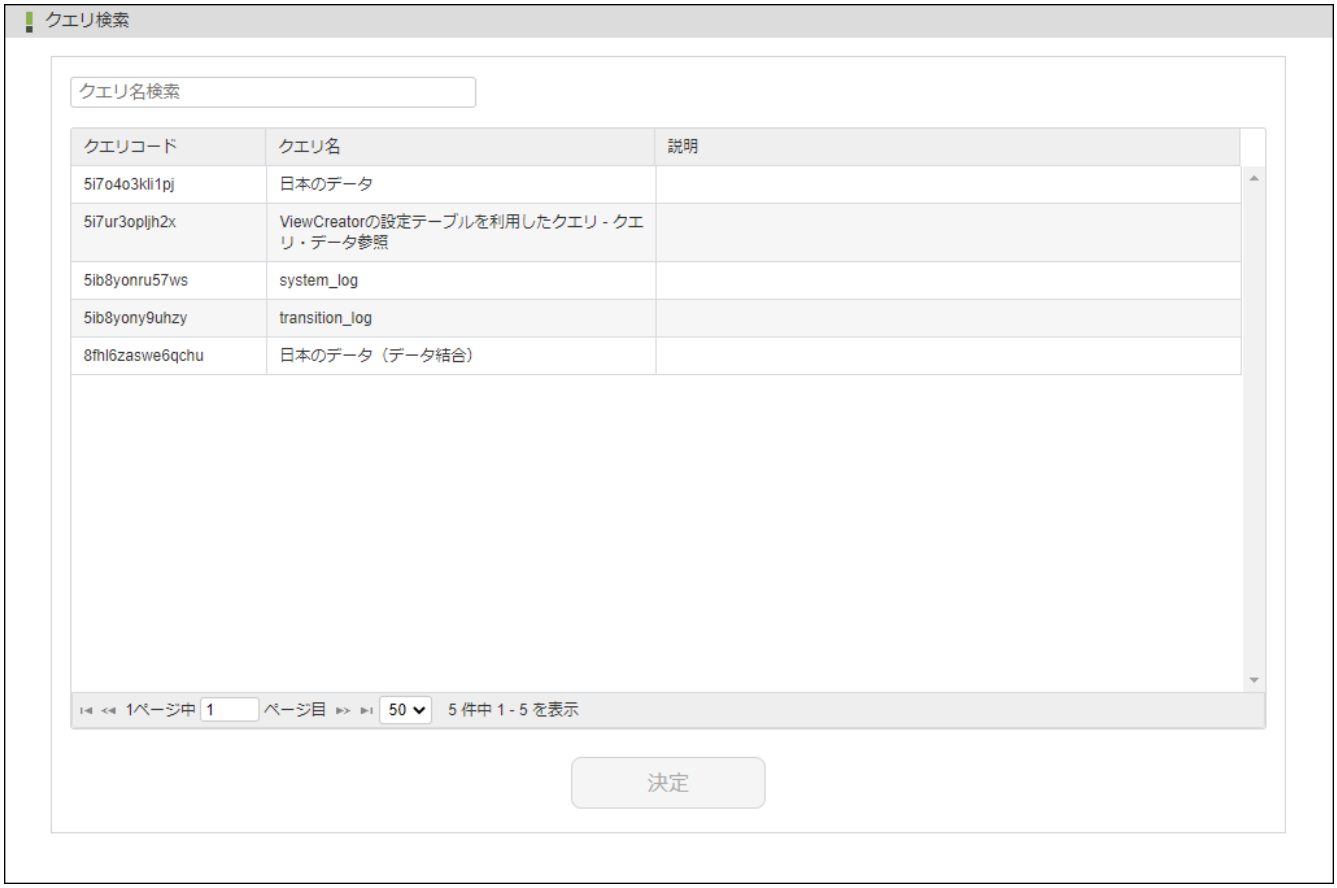
- ルーティングURLに指定できる文字列は「a-z」、「A-Z」、「0-9」、「-（ハイフン）」、「_（アンダースコア）」、「/（スラッシュ）」、「.（ピリオド）」のいずれかです。
- ルーティングURLを「/（スラッシュ）」から始めることはできません。

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。

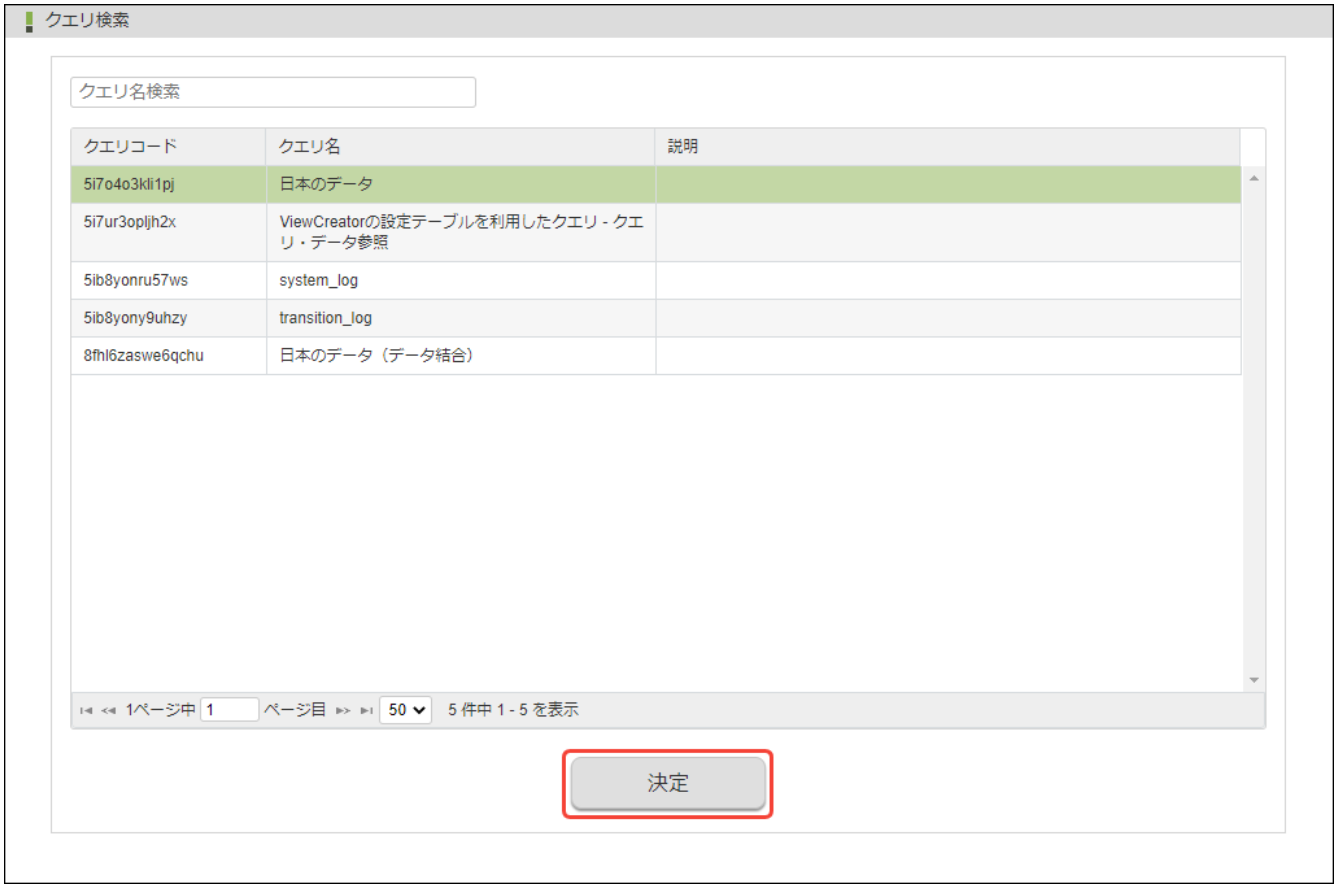


図：「クエリ検索」画面

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。

2. 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。

コラム

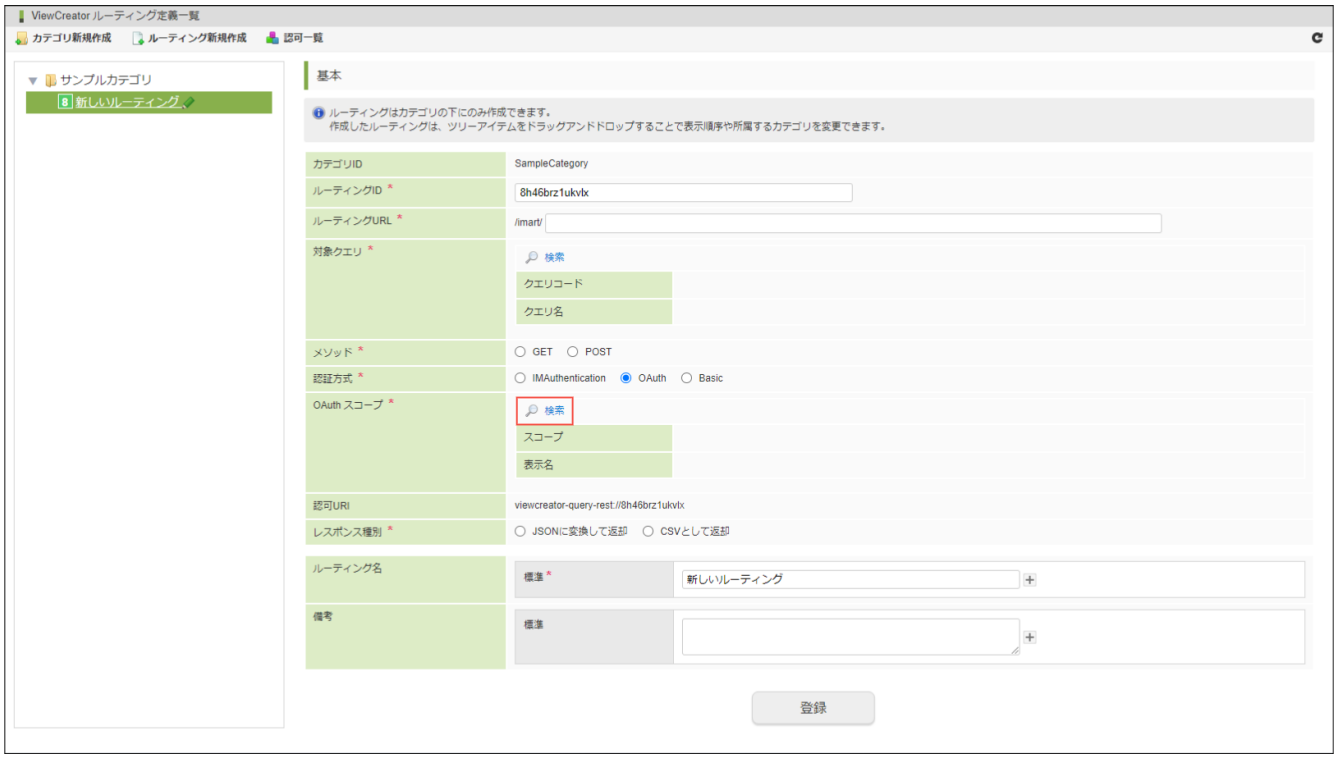
バーチャルテナントによる複数テナントを利用している環境において、リクエスト情報を利用したテナント自動解決機能を利用していない場合のみ、Basic認証のユーザ名にテナントを指定できます。
上記環境におけるBasic認証のユーザ名の指定方法による挙動の違いは以下の通りです。

- 「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 OAuth スコープ設定

OAuth スコープは「スコープ検索」画面で一覧から選択できます。

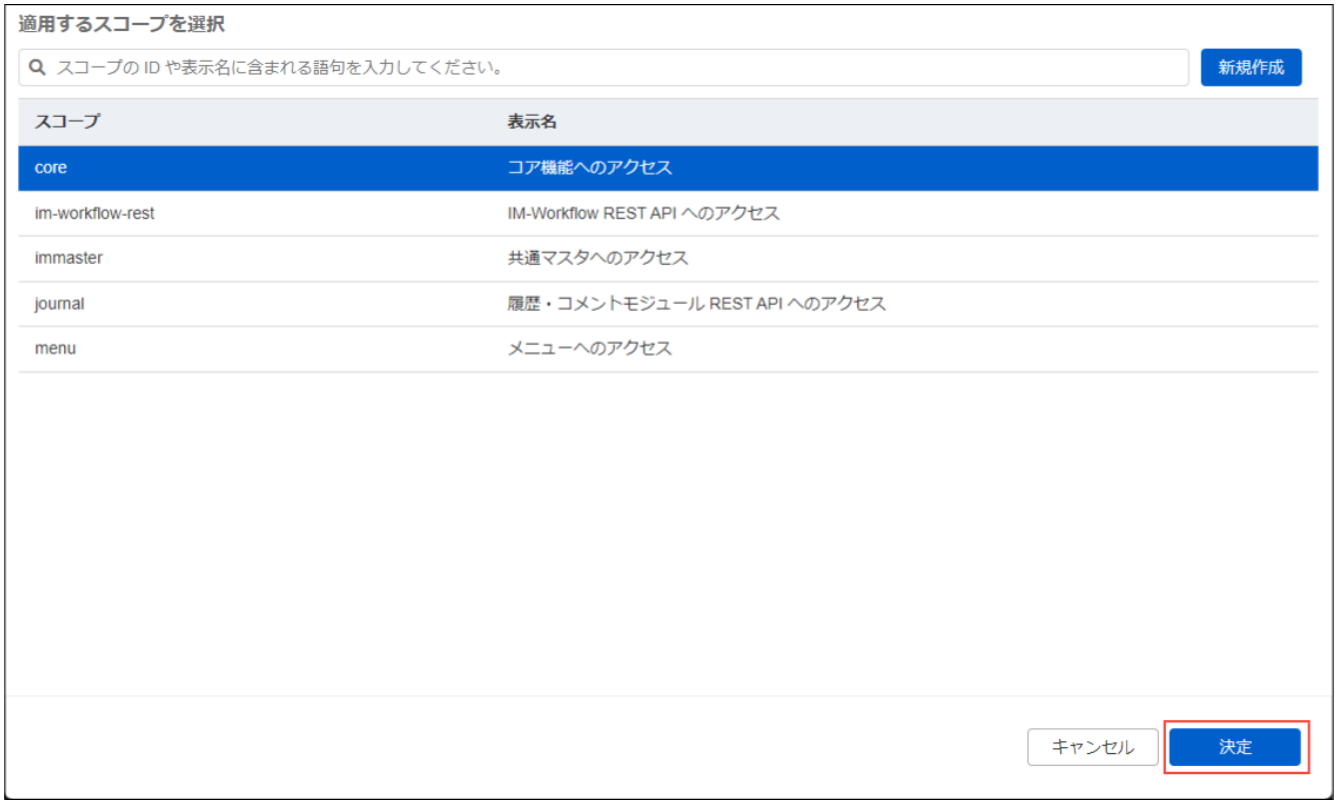
1. ルーティング情報の OAuth スコープの「検索」アイコンをクリックすると、「スコープ検索」画面が表示されます。



図：ルーティング情報 - OAuth スコープ

図：「スコープ検索」画面

2. OAuth スコープを選択して「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：「スコープ検索」画面 - スコープ選択

> **コラム**
>
> 「スコープ検索」画面から「新規作成」をクリックすることで、新規の OAuth スコープを作成できます。
> 詳細は 「 OAuth 管理者操作ガイド 」 - 「 スコープの登録 」 を参照してください。

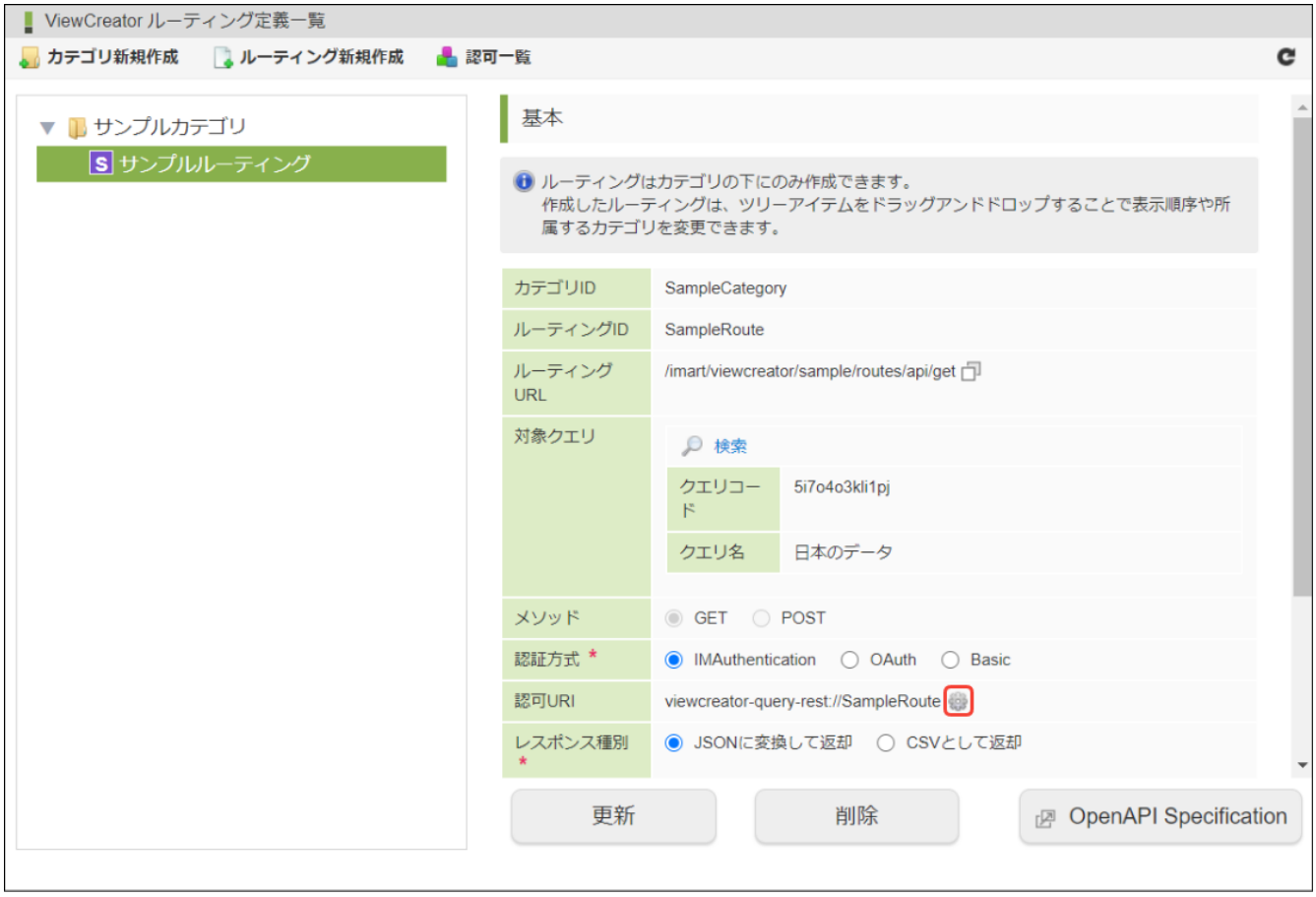
ルーティングの認可設定

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。

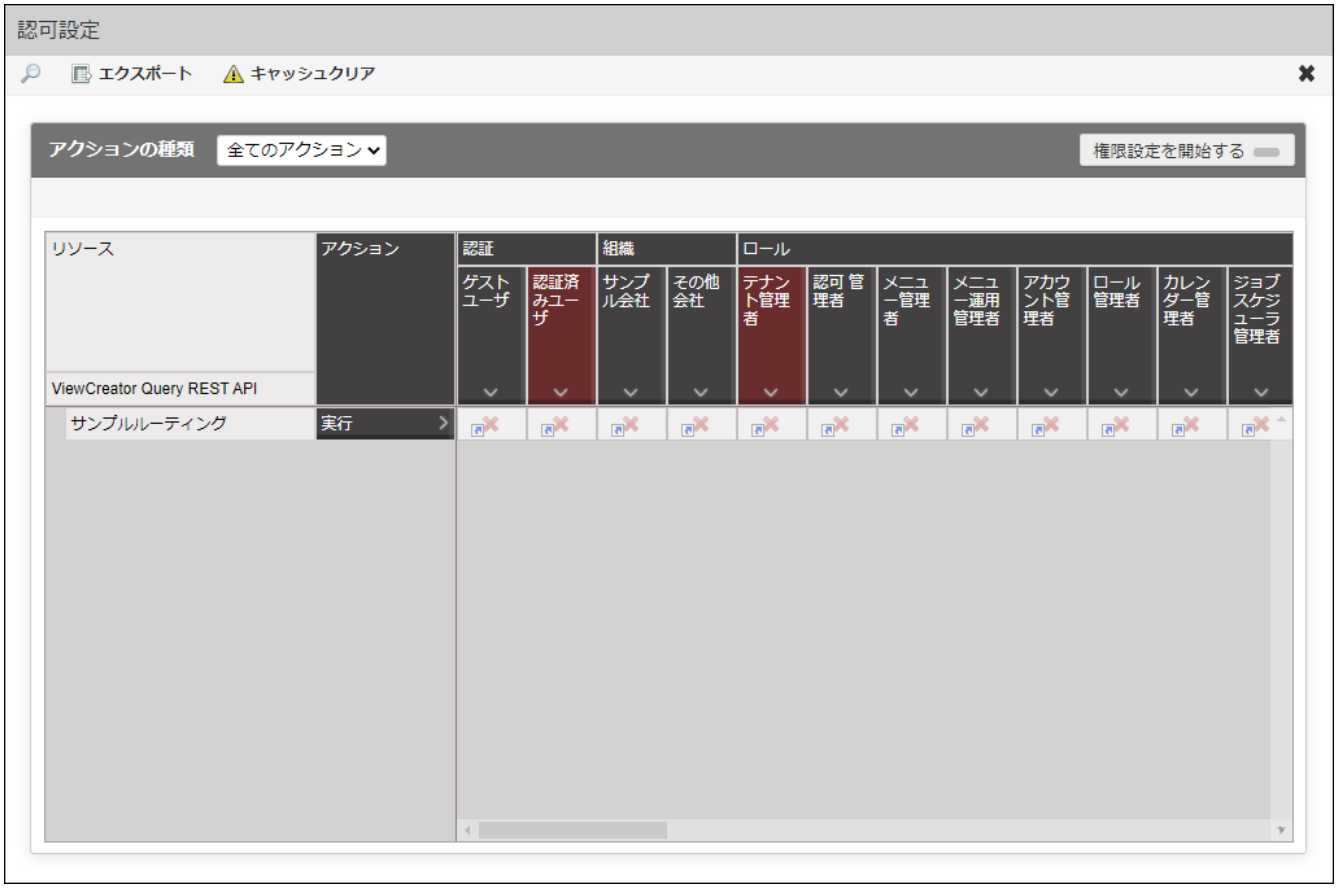
1. 認可設定をするルーティングをルーティングツリーから選択します。

2. 「認可URI」の「認可設定」アイコンをクリックします。



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

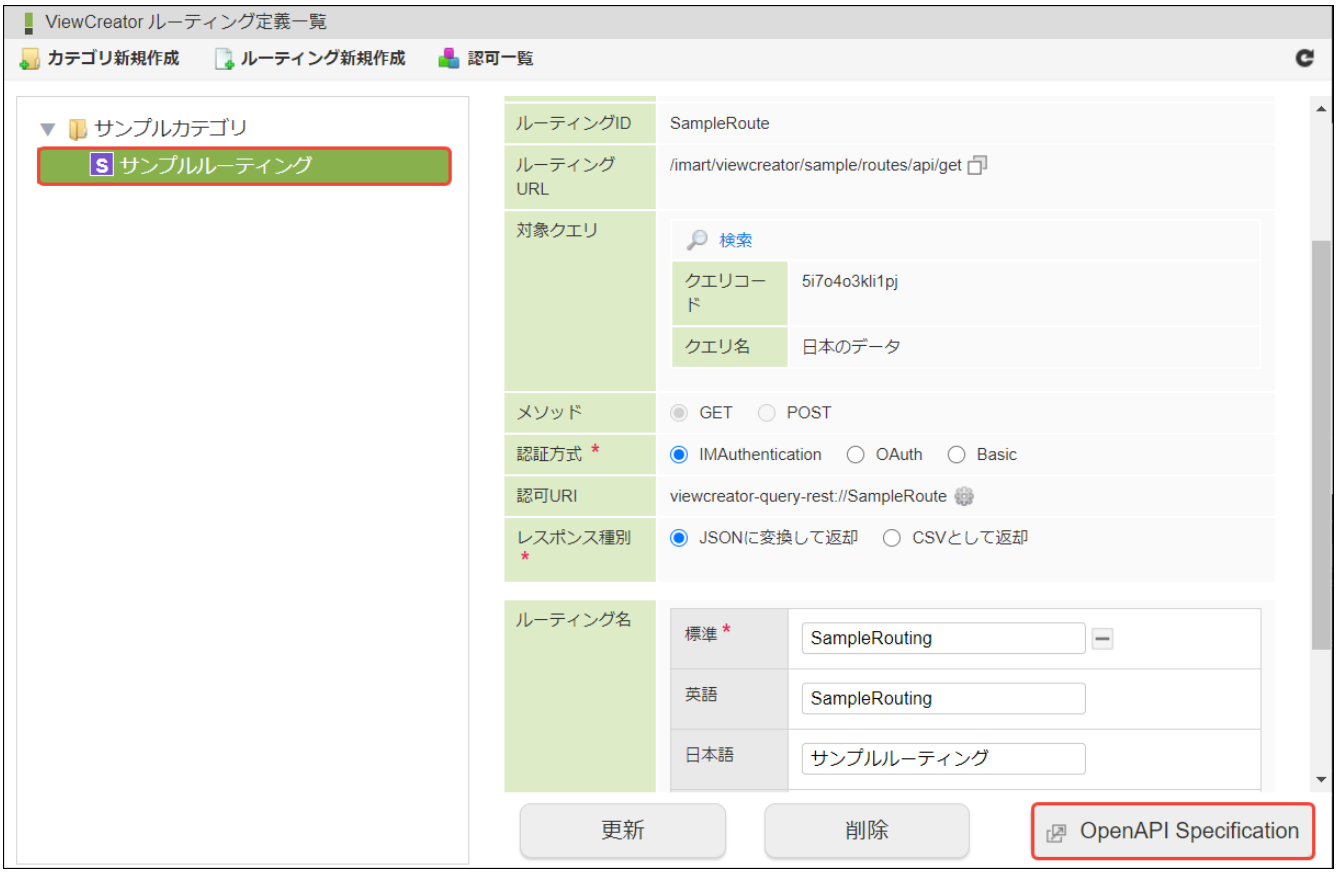
4. 認可設定を行います。認可の設定方法は「intra-mart Accel Platform 認可仕様書」の「認可設定画面」を参照してください。

> **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

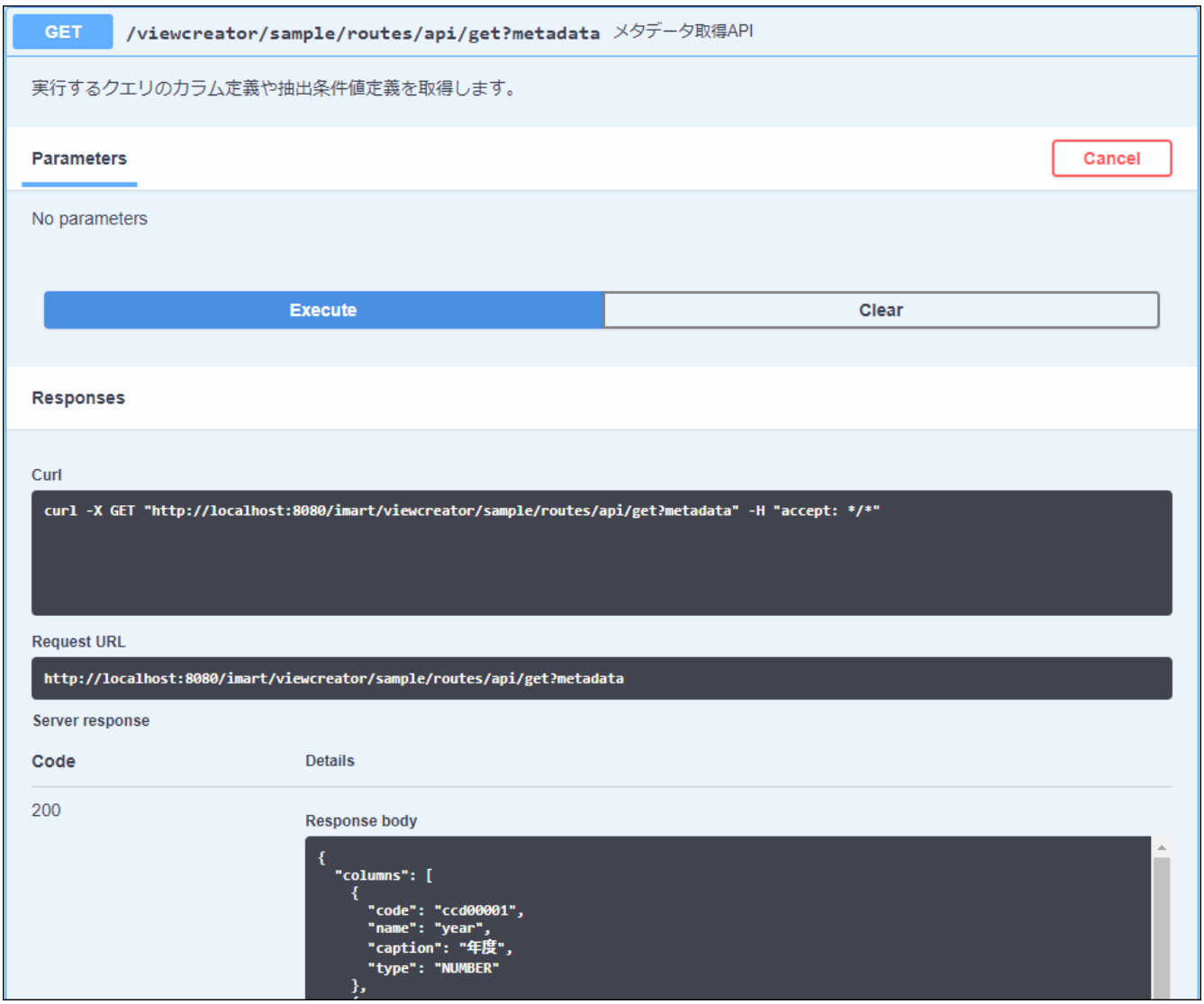
作成したルーティングは、OpenAPIを利用することで実行できます。
クエリの実行はREST APIを介して実行されます。
REST APIの詳細は「*REST API 仕様*」を参照してください。

- 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。

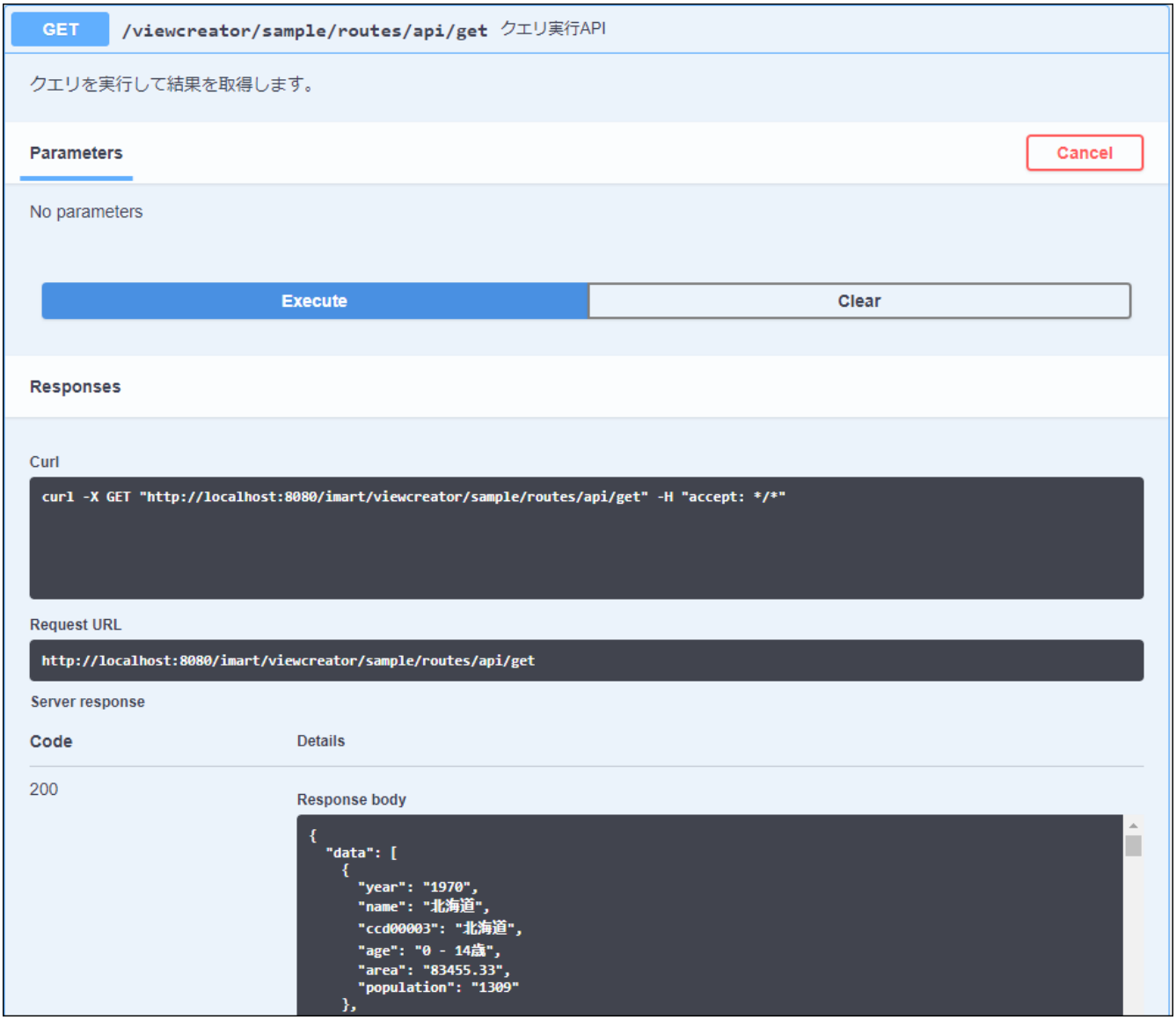


図：「ルーティング定義一覧」画面 - OpenAPIを開く

図：OpenAPI REST APIビューア



図：REST API実行 - クエリのメタ情報取得



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
limit	数値型	最大取得レコード件数を設定します。 省略した場合は、デフォルト設定（100件）が適用されます。 0を設定した場合は全件取得します。 設定例: *limit=10*

パラメータ	データ型	説明
offset	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: *offset=10*
columns	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: *columns=user_cd,ccd00003,user_name*
sort	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する場合に設定します。 *order* 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: *sort=user_cd&order=asc* コラム *order* 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。
order	文字列型	第１ソートキー（*sort*）に対するソート方向を設定します。 *asc* または *desc* を設定します。 設定例: *sort=user_cd&order=desc*
secondSort	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 *sort* パラメータの説明を参照してください。
secondOrder	文字列型	第２ソートキー（*secondSort*）に対するソート方向を設定します。 詳細は *order* パラメータの説明を参照してください。
thirdSort	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 *sort* パラメータの説明を参照してください。
thirdOrder	文字列型	第３ソートキー（*thirdSort*）に対するソート方向を設定します。 詳細は *order* パラメータの説明を参照してください。
responseKeyType	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。 ▪ *NAME* 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。 ▪ *CODE* カラムコードをレスポンスデータのキーに設定します。 ▪ *CAPTION* 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。 省略した場合は、デフォルト設定（*NAME*）が適用されます。 設定例: *responseKeyType=NAME* 注意 *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。
metadata	－	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「*レスポンスボディ*」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
x-jp-co-intra-mart-viewcreator-total-count	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。 *limit* パラメータや *offset* 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
}
```

data	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。
totalCount	実行結果の総件数です。 レスポンスヘッダに格納される *x-jp-co-intra-mart-viewcreator-total-count* 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"
   }
 ]
}
```

columns	クエリ定義に含まれるカラム情報の一覧です。 ▪ *code* *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。 ▪ *name* DDL上で定義されたフィールドの物理名です。 ▪ *type* 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。 ▪ *caption* 「クエリ編集」画面で設定したキャプションです。
conditions	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。 ▪ *key* パラメータ名（*KEY_NAME*）が格納されます。 ▪ *type* 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。

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（JVM 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

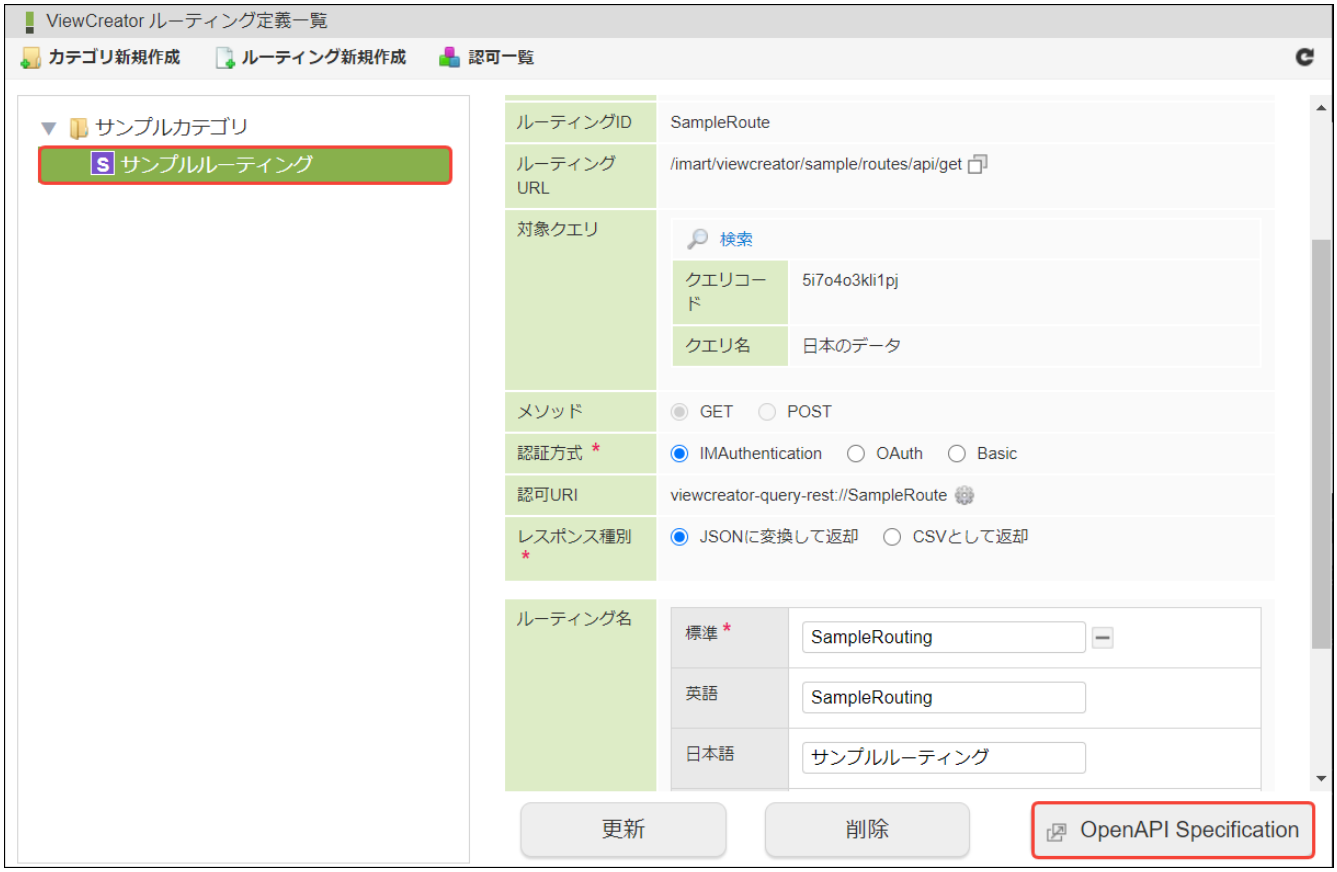
OpenAPIはソースコードやドキュメントなどにアクセスせずに、APIの機能を理解できるREST APIの標準インタフェースを定義しています。
OpenAPI仕様はREST APIを記述するためのフォーマットです。

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

1. 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。



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

2. OpenAPIの提供するREST APIビューア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ビューア

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。

3. 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

4. 「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。

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「リクエストパラメータ」を参照してください。 ▪ パラメータ名、説明、タイプ ▪ データ形式、デフォルト値
「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*

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	Response body


```
{
  "data": [
    {
      "age": "0 - 14歳",
      "year": "1970",
      "area": "1862.01"
    },
    {
      "age": "0 - 14歳",
      "year": "1970",
      "area": "1893.76"
    }
  ],
  "totalCount": 987
}
```


Download

Response headers

```
content-length: 130
content-type: application/json
date: Tue02 Mar 2021 01:00:44 GMT
server: Resin/4.0.64
x-jp-co-intra-mart-viewcreator-total-count: 987
```

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「*レスポンスボディ*」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「*レスポンスヘッダ*」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。

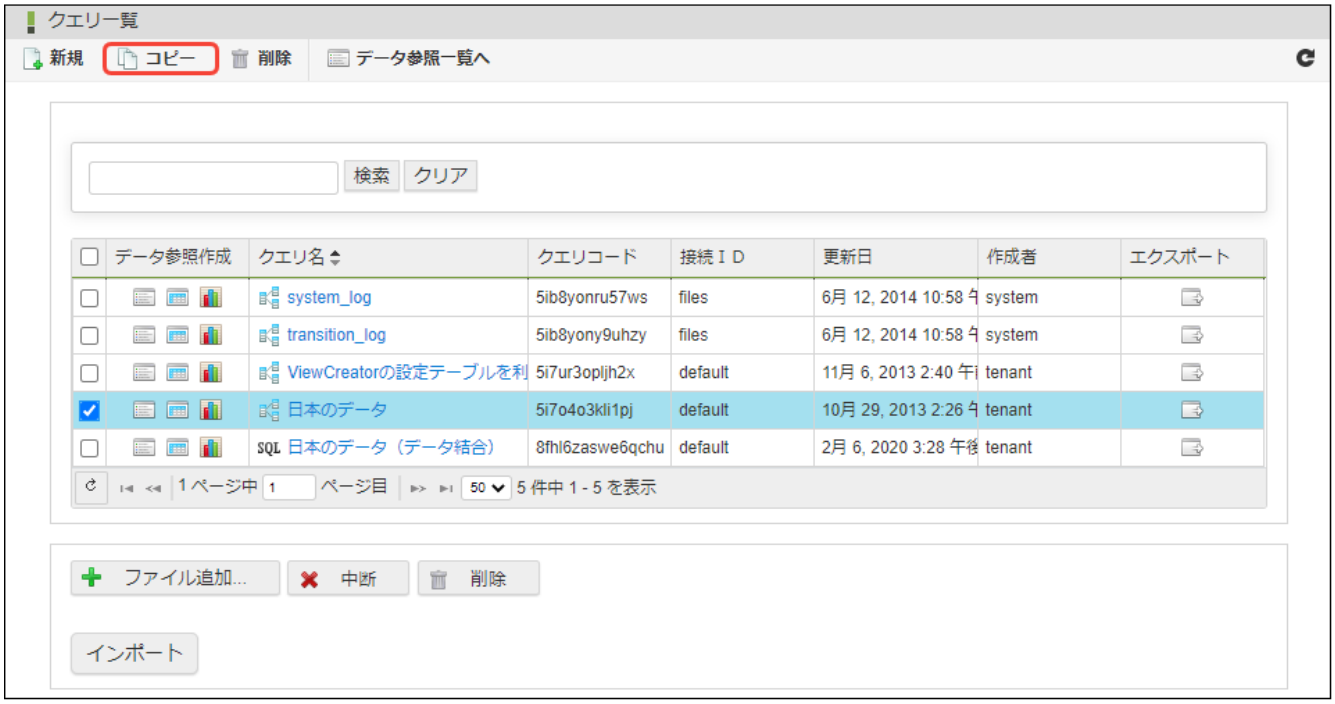
図：完成イメージ

> **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

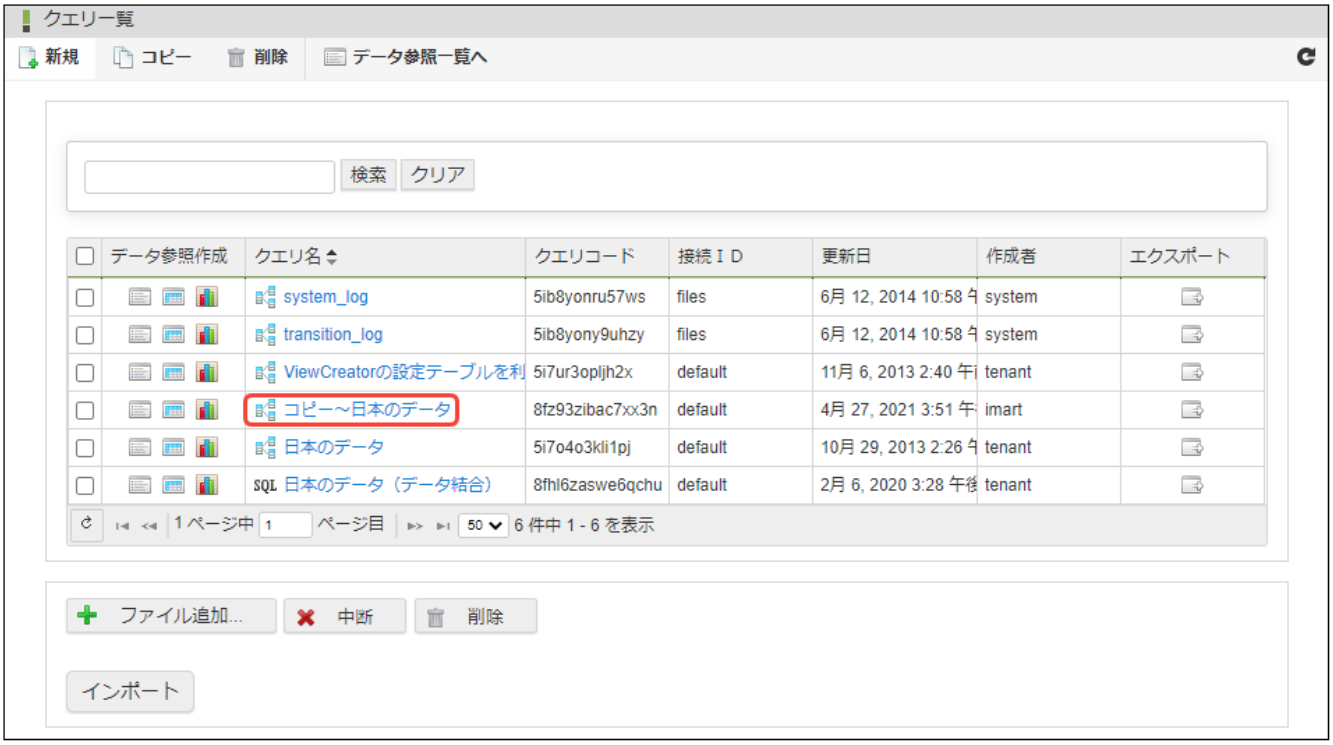
はじめに、実行するクエリを作成します。
本作成例では、サンプルデータのクエリを利用します。

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：「クエリ一覧」画面 - クエリのコピー

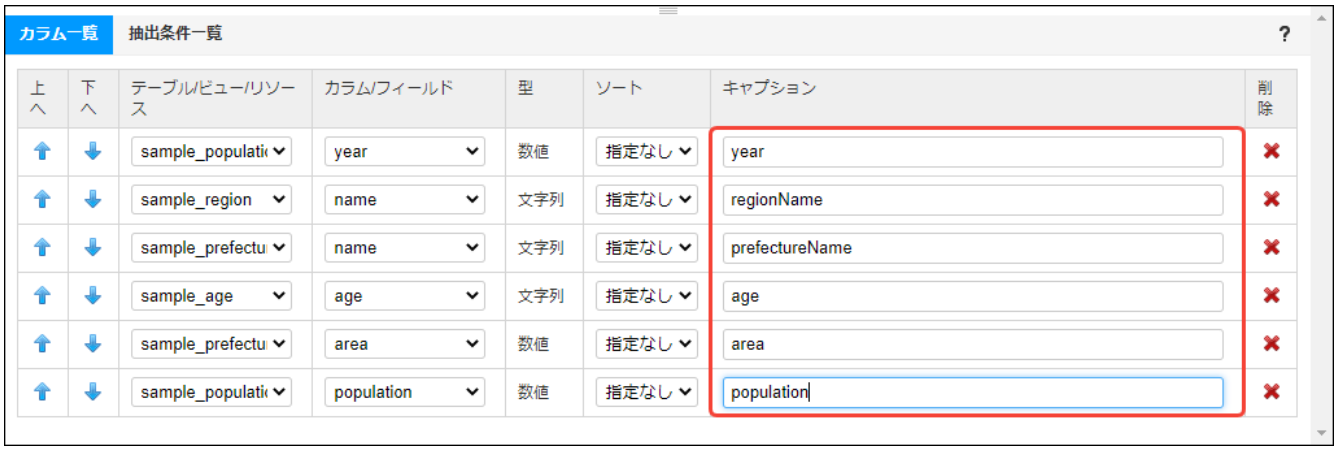
3. 一覧からコピーしたクエリ「コピー～日本のデータ」をクリックし、「クエリ編集」画面を表示します。



図：「クエリ一覧」画面 - 「コピー～日本のデータ」

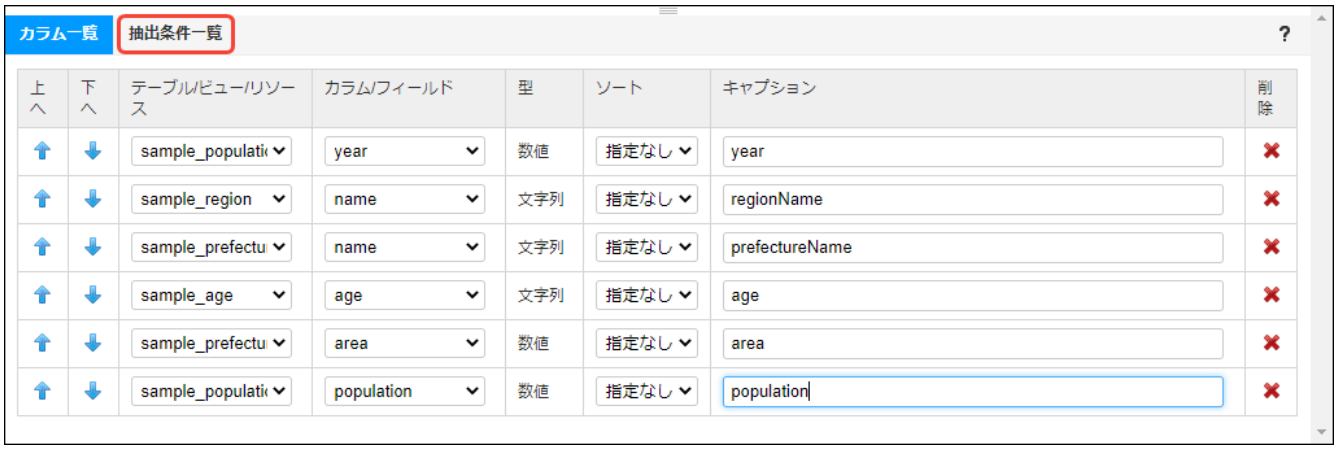
4. 「クエリ編集」画面下部の「カラム一覧」タブの「キャプション」を以下のように編集します。

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*



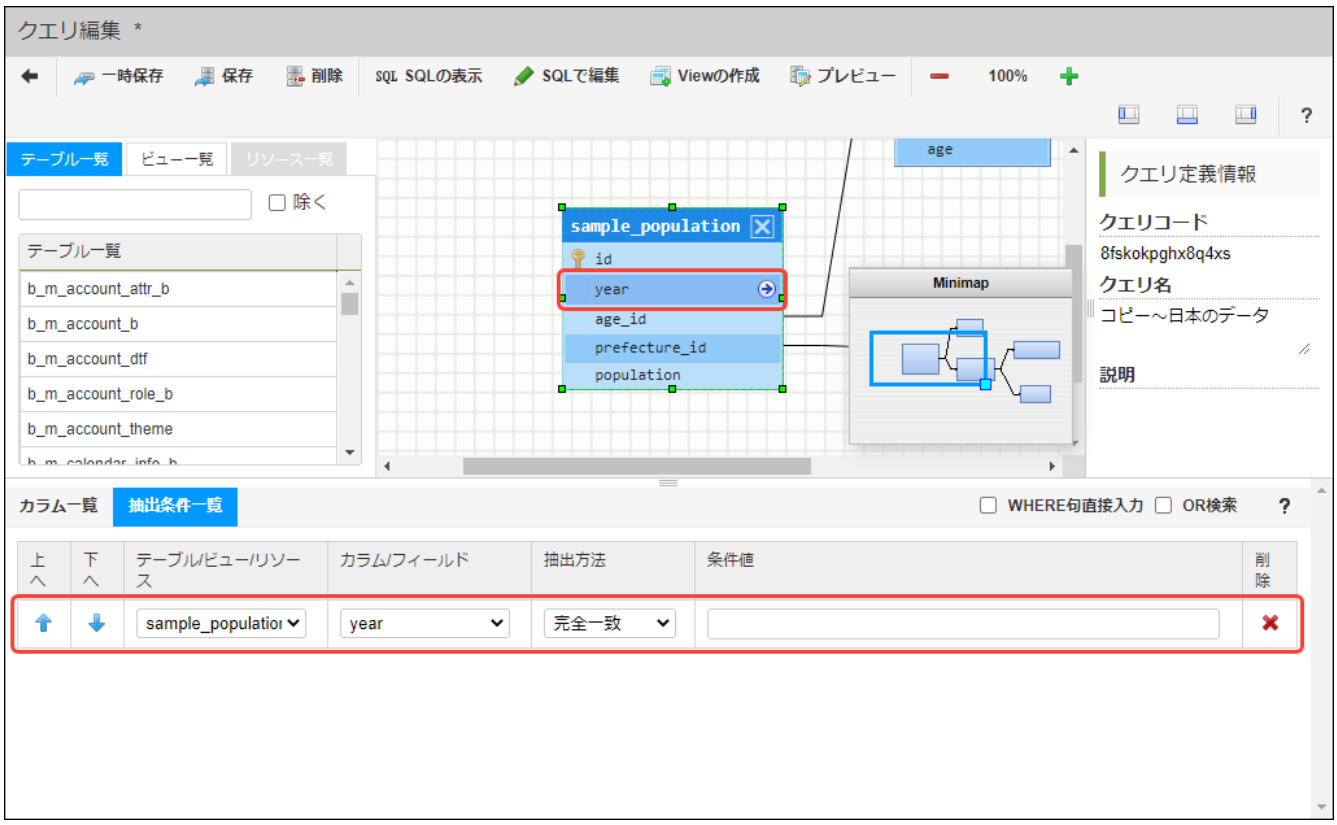
図：「クエリ一覧」画面 - キャプション設定

5. 「クエリ編集」画面下部の「抽出条件一覧」タブをクリックします。



図：「クエリ編集」画面 - 「抽出条件一覧」タブ

6. 「クエリ編集」画面のデザイナで「*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）。



図：「クエリ編集」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)%>*

図：「クエリ編集」画面 - 抽出条件設定

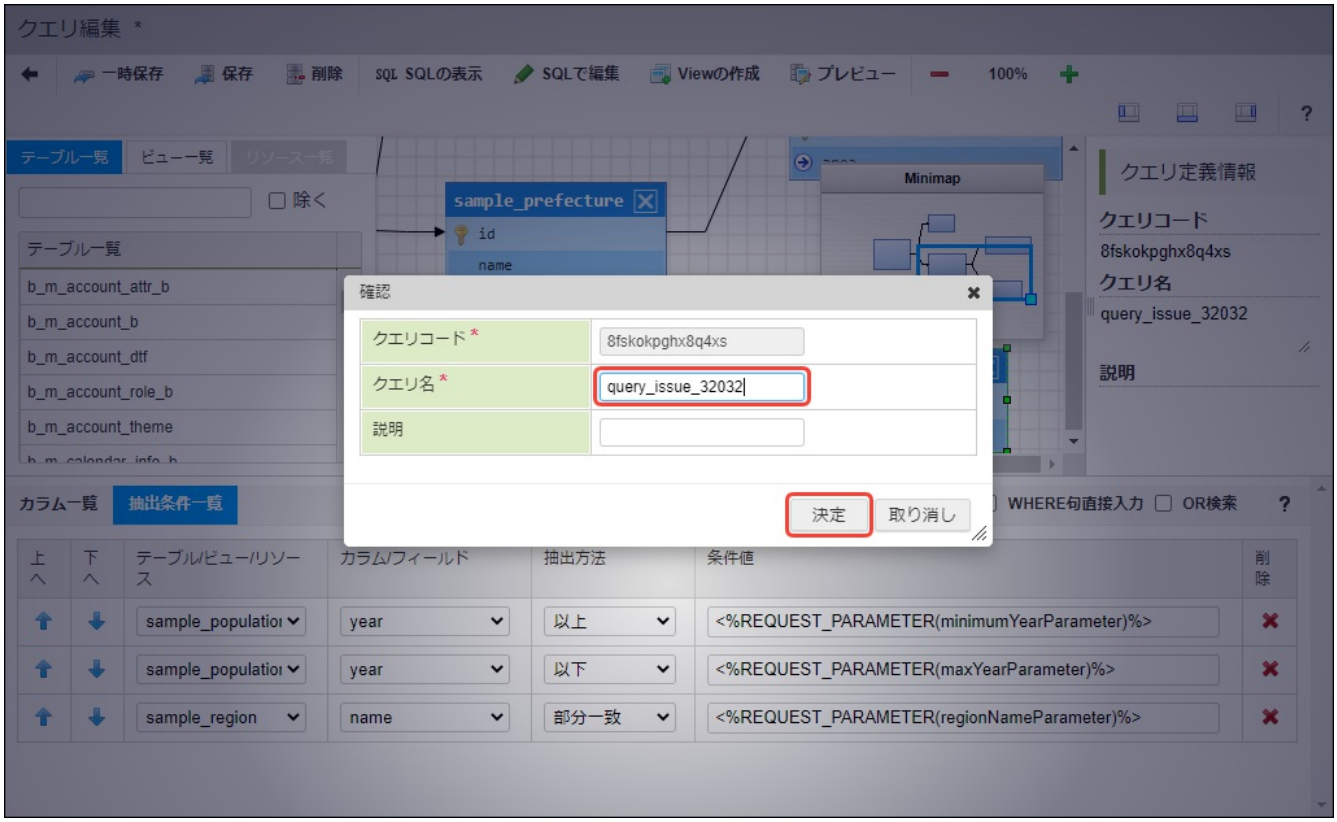
8. 同様に、以下のとおりに抽出条件を追加します。
   - テーブル/ビュー/リソース：*sample_population*
   - カラム/フィールド：*year*
   - 抽出方法：「以下」
   - 条件値：*<%REQUEST_PARAMETER(maxYearParameter)%>*

   - 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)%>*

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。

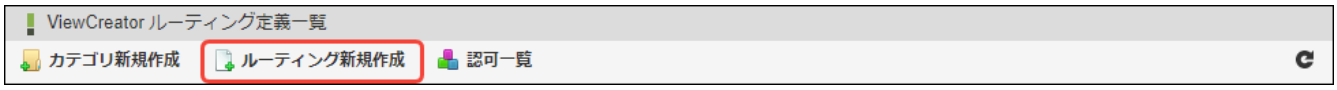


図：「クエリ編集」画面 - クエリ保存

ルーティング定義の作成

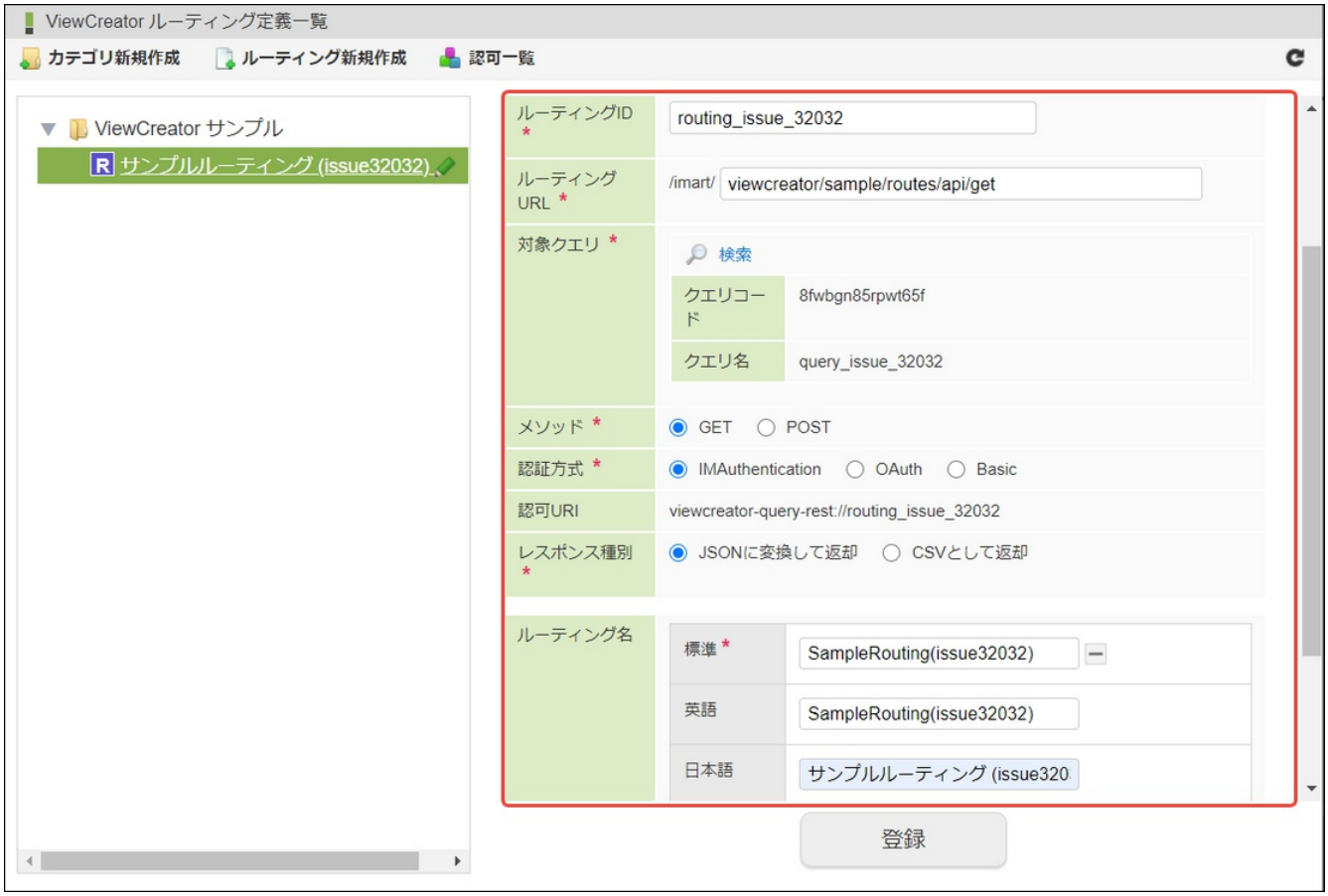
次に、ルーティング定義を作成します。

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)*



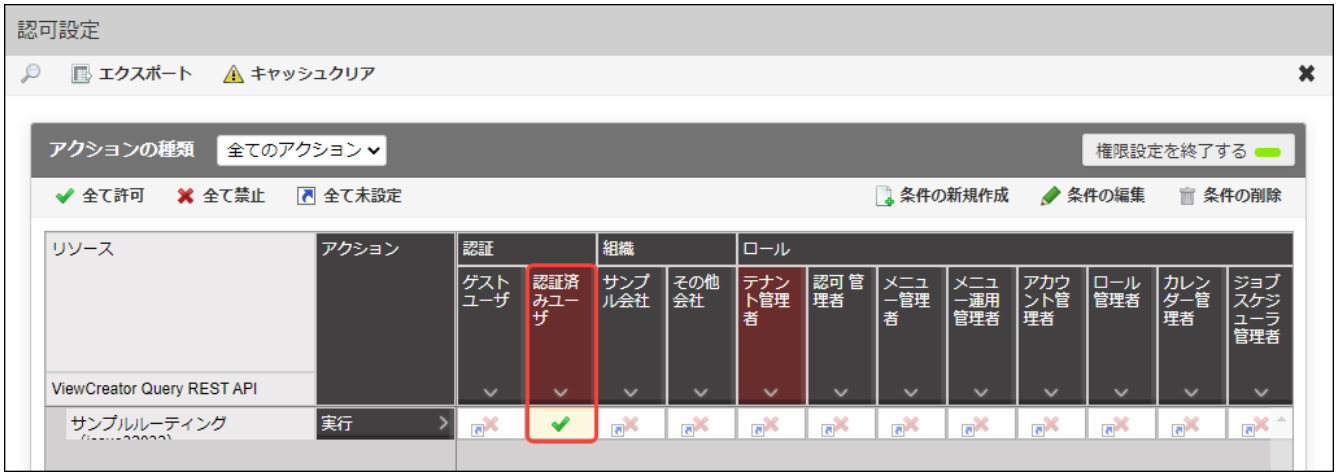
図：「ルーティング定義一覧」画面 - ルーティング情報の入力

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**

> **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。

   > **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)*

   > **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。

> 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

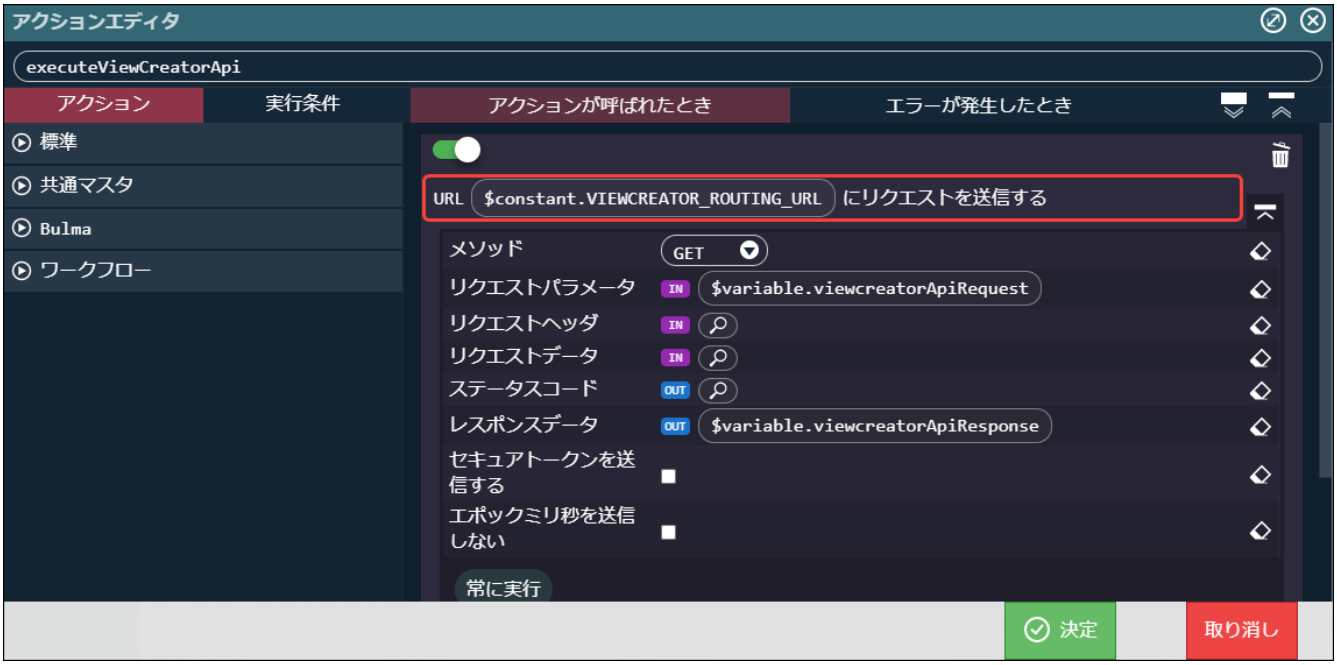
IM-BloomMakerとの連携

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。

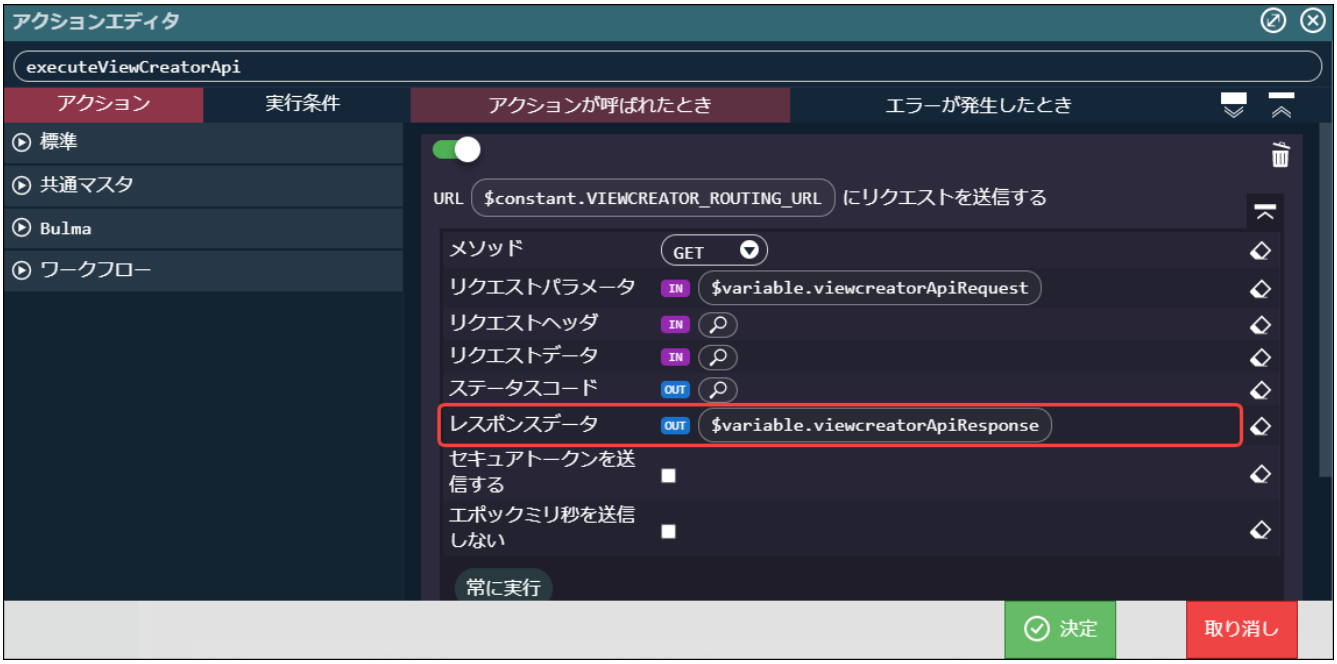
図：*executeViewCreatorApi* アクション - リクエストパラメータ

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
viewcreatorApiRequest.limit	最大取得レコード件数を設定する *limit* パラメータの値を保持します。
viewcreatorApiRequest.offset	レコードの取得開始位置を設定する *offset* パラメータの値を保持します。
viewcreatorApiRequest.columns	取得対象とするカラムを設定する *columns* パラメータの値を保持します。
viewcreatorApiRequest.sort	第１ソートキーを設定する *sort* パラメータの値を保持します。
viewcreatorApiRequest.order	第１ソートキーに対するソート方向を設定する *order* パラメータの値を保持します。
viewcreatorApiRequest.secondSort	第２ソートキーを設定する *secondSort* パラメータの値を保持します。
viewcreatorApiRequest.secondOrder	第２ソートキーに対するソート方向を設定する *secondOrder* パラメータの値を保持します。
viewcreatorApiRequest.thirdSort	第３ソートキーを設定する *thirdSort* パラメータの値を保持します。
viewcreatorApiRequest.thirdOrder	第３ソートキーに対するソート方向を設定する *thirdOrder* パラメータの値を保持します。
viewcreatorApiRequest.responseKeyType	レスポンスデータのキー項目を設定する *responseKeyType* パラメータの値を保持します。
viewcreatorApiRequest.minimumYearParameter	クエリの抽出条件（動的パラメータ *<%REQUEST_PARAMETER(minimumYearParameter)%>*）の値を保持します。
viewcreatorApiRequest.maxYearParameter	クエリの抽出条件（動的パラメータ *<%REQUEST_PARAMETER(maxYearParameter)%>*）の値を保持します。
viewcreatorApiRequest.regionNameParameter	クエリの抽出条件（動的パラメータ *<%REQUEST_PARAMETER(regionNameParameter)%>*）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
viewcreatorApiResponse.data	REST APIのレスポンスボディの *data* 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、*responseKeyType* パラメータに「*CAPTION*」を設定しているので、カラムの論理名（*year*、*regionName*、*prefectureName*、*age*、*area*、*population*）を使用します。
viewcreatorApiResponse.totalCount	REST APIのレスポンスボディの *totalCount* 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（*data* パラメータ）と関連付けるために、*list* プロパティに変数 *viewcreatorApiResponse.data* 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、*textContent* プロパティに各カラムデータの格納先変数（*year*、*regionName*、*prefectureName*、*age*、*area*、*population*）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（*totalCount* 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 *total* プロパティに *viewcreatorApiResponse.totalCount* を設定します。

注意事項

データ参照表示時のリソースの使用量増加について

データ参照を表示する際の表作成や集計・計算処理で大量のデータを扱う必要が発生した場合、リソースの使用量が急増してパフォーマンスの劣化やメモリの枯渇等が発生する可能性があります。
そのため、一度に大量のデータを取得するクエリを実行する場合や大量の選択肢を持つ検索条件を作成する場合は、リソースの使用量に注意する必要があります。

「検索条件値・選択肢リスト」に「データ参照を利用」を選択している場合

「検索条件値・選択肢リスト」に「データ参照を利用」を選択している場合、選択肢を作成する際に指定されたデータ参照の全データを走査する必要があるため、より多くのリソースを使用します。
「ViewCreatorの設定」-「検索設定」を設定することで、検索条件に利用するデータ参照のリソースの使用量を抑制できます。

> コラム
>
> 「検索設定」は 2022 Spring(Eustoma) から設定できます。
> 「検索条件値・選択肢リスト」については「*検索の条件値入力に選択型コンポーネントを利用する*」を参照してください。

表示するデータ参照がクロス集計の場合

クロス集計で大量のデータを集計する場合、多くのリソースを使用する可能性があります。
「クロス集計で大量のデータを扱えるようにします。」にて、クロス集計表作成時のリソースの使用量を抑制する設定が追加されました。
「ViewCreatorの設定」-「クロス集計設定」により、データ取得時に適用されるフェッチサイズ（1度の通信で取得するレコード数）や集計表の行数および列数の上限を設定できます。

> コラム
>
> 「クロス集計設定」は 2022 Spring(Eustoma) から設定できます。

クエリの取得対象にバイナリ型データが含まれている場合

クエリで取得するデータにバイナリ型のフィールドが含まれている場合、フィールドに格納されているデータのサイズによっては大量のリソースを必要とします。
「バイナリ型のフィールドからデータを取得しないようにする設定を追加します。」にて、バイナリ型のフィールドからデータを取得しないようにする設定が追加されました。
「ViewCreatorの設定」-「バイナリ型カラムの利用」により、バイナリ型のフィールドのデータ取得について有効・無効を設定できます。

> コラム
>
> 「バイナリ型カラムの利用」は 2022 Spring(Eustoma) から設定できます。