



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
 - 4.20. ViewCreator SQLビルダ アシスタント によるSQL生成
- 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
 - 5.2. アクセス種別-認可の利用について
- 6. 付録
 - 6.1. チュートリアル
 - 6.2. ViewCreator SQLビルダ アシスタント
 - 6.3. ログ関連

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 <ul style="list-style-type: none"> ▪ 「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 <ul style="list-style-type: none"> ▪ 「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 <ul style="list-style-type: none"> ▪ 「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 <ul style="list-style-type: none"> ▪ 「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 <ul style="list-style-type: none"> ▪ 「利用可能な関数一覧」にFORMAT関数へのロケルの指定についての記載を追加 ▪ 「参照権の設定」にゲストユーザーと認証済みユーザー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カラムを追加 ▪ 「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 <ul style="list-style-type: none"> ▪ 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	<p>第8版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 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	<p>第9版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 「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	<p>第10版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 「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	<p>第11版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 「クロス集計」の画面インター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	<p>第12版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 「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	<p>第13版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 「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	<p>第14版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> ■ 「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	<p>第15版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> ■ 「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	<p>第16版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> ▪ 「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	<p>第17版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> ▪ 「設定可能なパラメータ」のdest_storage_pathの説明を修正
2018-04-01	<p>第18版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> ▪ 「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	<p>第19版 下記を追加・変更しました。</p> <ul style="list-style-type: none"> ▪ 「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	<p>第20版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第21版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第22版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第23版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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 ▪ 「クエリー一覧画面」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」のクエリー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「クエリー一覧に表示されるクエリについて」を追加
2020-08-01	<p>第24版 下記を変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第25版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第26版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第27版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第28版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第29版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ▪ 「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	<p>第30版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 「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	<p>第31版 下記を追加・変更しました</p> <ul style="list-style-type: none"> ■ 「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	<p>第32版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「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
2024-06-28	<p>第33版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「付録」および「チュートリアル」を追加 ■ 「チュートリアル」に「IM-LogicDesigner を利用して取得したタイムスタンプ型のレコードを ViewCreator のデータ参照のリスト集計で表示する。」を追加
2024-10-01	<p>第34版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「グラフ集計の作成」に「ECharts」を追加 ■ 「作成したデータ参照の閲覧」と「国際化とヘッダ・フッタ」にCSV出力したファイルのカラムヘッダの値についての説明を修正 ■ 「動的パラメータ」に<%TIME_ZONE_ID%>の説明を追加 ■ 「クエリ一覧画面」と「データ参照の作成」のコラムに、XMLファイルのフォーマット変更に関する記述を追加 ■ 「クエリとデータ参照のインポート/エクスポート」の廃止に伴い、関連する記述を削除 ■ Flash版のクエリ編集画面の廃止に伴い、「クエリの作成」のコラムを修正 ■ 「ViewCreator SQLビルダ アシスタントによるSQL生成」を追加 ■ 「ViewCreator全体の権限設定」に「ViewCreator SQLビルダ アシスタントを利用する場合」を追加 ■ 「付録」に「ViewCreator SQLビルダ アシスタント」を追加
2024-11-29	<p>第35版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「ViewCreator SQLビルダ アシスタントによるSQL生成」のエディションについてのコラムを修正

変更年月日	変更内容
2025-04-01	<p>第36版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 2025 Spring(Kamille)からは、末尾にセミコロンや行コメントを含んだSQLがSQLビルダで使用可能になったため、「ViewCreator SQLビルダ アシスタント の利用例」の注意を削除しました ■ 「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「設定方法」を追加 ■ 「認可の設定」の管理者リソースの説明を修正 ■ 「ViewCreator全体の権限設定」に「アクセス種別-認可を利用する場合」を追加 ■ 「データ参照の作成」に「権限設定」を追加 ■ 「参照権の設定」を「権限設定」の配下に移動 ■ 「データ参照の作成」に「認可の設定」を追加 ■ 「注意事項」に「アクセス種別-認可の利用について」を追加 ■ デザインシステムに準拠したリスト集計画面の提供に伴い「データ参照の作成」の画像と内容について変更および追加 ■ 「帳票出力の設定」から wkhtmltopdf の設定についての記述を削除 ■ 「利用可能な関数一覧」に、DATAVIEWNAME 関数と DATAVIEWCODE 関数の説明を追加 ■ 「インポート・エクスポート」に異なるデータベース製品・データベースバージョン間のインポート・エクスポートについての注意を追加 ■ 「動的パラメータを利用可能な個所」の、シングルクォーテーションの扱いについての説明を修正
2025-04-25	<p>第37版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「ViewCreator SQLビルダ アシスタントによるSQL生成」のエディションについてのコラムを修正
2025-05-30	<p>第38版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「リスト集計の作成」のタイムゾーン設定の表示に関する注意事項を修正
2025-08-29	<p>第39版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「認可の設定」のコラム内のアクセス種別-認可に関するURLを修正
2025-10-01	<p>第40版 下記を追加しました</p> <ul style="list-style-type: none"> ■ 「検索タイプについて」に利用可能な型の項目を追加 ■ 「ViewCreator 操作ログについて」を追加 ■ 「ViewCreator全体の権限設定」と「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」でテーブル単位のアクセス権設定に関する動作仕様の変更点を追加 ■ 「ViewCreator SQLビルダ アシスタントによるSQL生成」の「前提条件」に著作権等に関する注意事項を追加 ■ 「動的パラメータ」の説明を修正 ■ 新しい動的パラメータ「<code><%REQUEST_PARAMETER_VALUES(KEY_NAME)%></code>」と「<code><%REQUEST_PARAMETER_NUMBER_VALUES(KEY_NAME)%></code>」を追加 ■ 「付録」に「ViewCreator SQLビルダ アシスタント 利用時のコスト」を追加

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

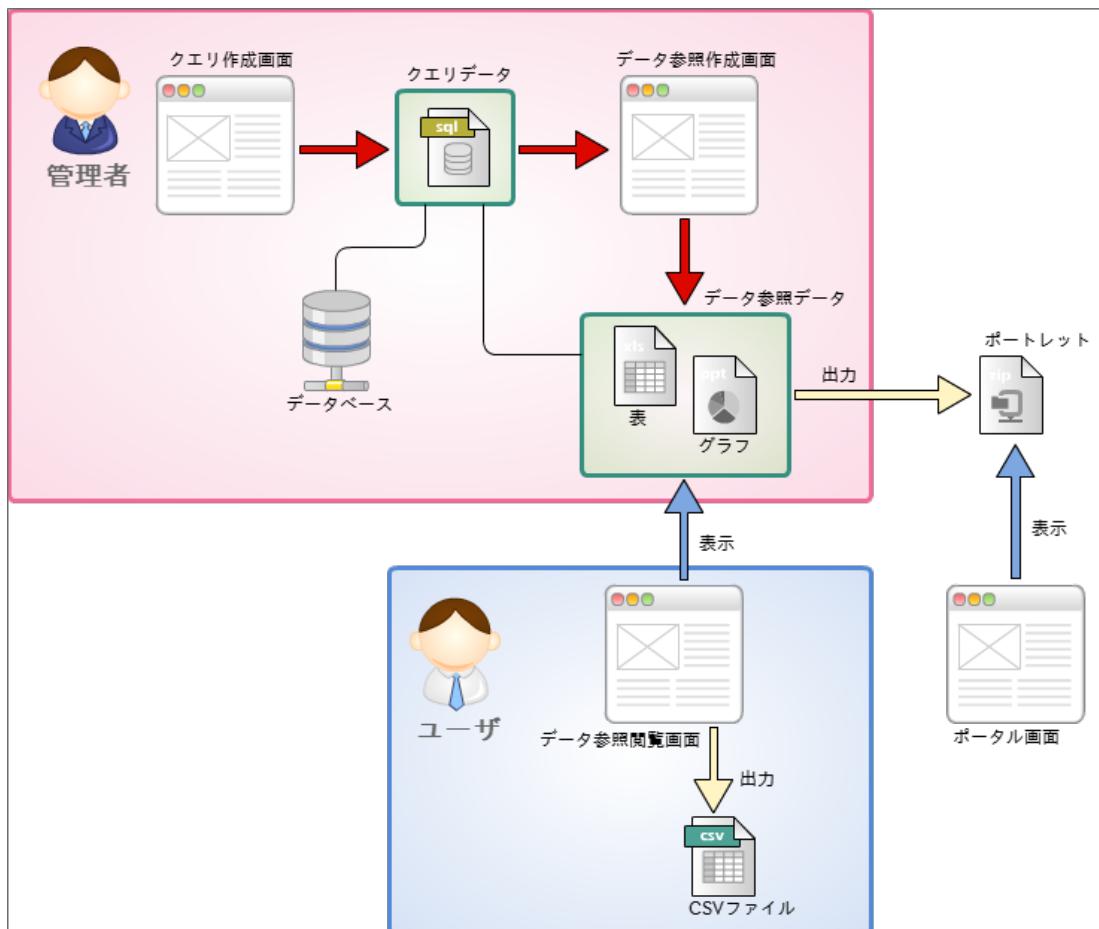
SQLの直接入力による作成は2020 Spring(Yorkshire)から利用可能です。

- 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）をさせることができます。

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

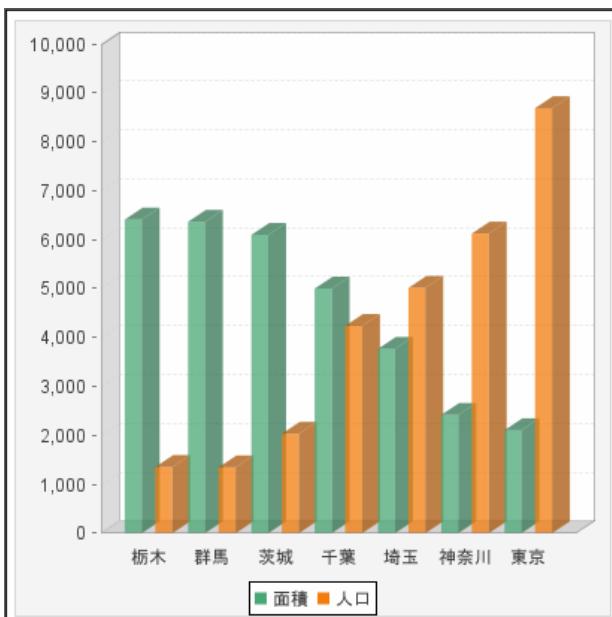
管理者はまず、既存のテーブル情報を用いてクエリを作成します。

クエリを作成したら、それを元にデータ参照を作ります。

[Top](#) > [1970](#) > 近畿

年度	地域名	都道府県名	年齢	人口(合計)	
		+ 京都	-	2,250	
		+ 三重	-	1,542	
		+ 滋賀	-	889	
1970	近畿	+ 大阪	-	7,621	★
		+ 奈良	-	930	
		+ 兵庫	-	4,668	
		+ 和歌山	-	1,042	
				18,942	

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	タイムスタンプ				バイナリ型	真偽値型
	文字列型	数値型	日付型	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

データベース				タイムスタンプ		
名	文字列型	数値型	日付型	型	バイナリ型	真偽値型
SQL Server	varchar char nvarchar nchar ntext	bigint int smallint tinyint decimal numeric money smallmoney float real	date	datetime smalldatetime datetime2	binary varbinary	bit
PostgreSQL	varchar character text	smallint bigint decimal numeric real	date	timestamp	bytea	boolean
SAP HANA	VARCHAR NVARCHAR CHAR NCHAR	BIGINT SMALLINT INTEGER DOUBLE DECIMAL TINYINT REAL	DATE	TIMESTAMP	BLOB CLOB NCLOB	BOOLEAN

コラム

真偽値型は2014 Summer(Honoka)から利用可能です。

注意

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

注意

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

ここでは基本的な操作について紹介します。

ViewCreator全体の権限設定

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

目次

- 認可の設定
 - ViewCreator SQLビルダ アシスタントを利用する場合
 - アクセス種別-認可を利用する場合
- メニューの権限設定

認可の設定

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

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

リソース名	説明
データ参照の閲覧・作成	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する設定です。ViewCreator管理者ロールをユーザに付与して使用します。

リソース名	説明
インポート・エクスポート	<p>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。</p> <p>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。</p> <ul style="list-style-type: none"> ・ 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 <p>個別でインポート/エクスポートする機能は管理外です。</p> <p>一般的には管理者ユーザのみを使用可能にする設定です。ViewCreator管理者ロールをユーザに付与して使用します。</p>
管理者	<p>他のユーザが作成したクエリやデータ参照に対して参照、変更や削除を可能にします。</p> <p>ただし各機能を利用するためには別途以下の認可設定が必要です。</p> <ul style="list-style-type: none"> ・ クエリ一覧、クエリ編集 「クエリの作成」への認可設定が必要です。 ・ データ参照一覧、データ参照編集 「データ参照の閲覧・作成」への認可設定が必要です。 ・ データ参照 アクセス種別-参照権の場合 「データ参照の閲覧・作成」への認可設定が必要です。 アクセス種別-認可の場合 各データ参照に対しての認可リソースへの認可設定が必要です。
クエリの作成	<p>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。</p> <p>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。</p> <p>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。</p>
Viewの作成	<p>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。</p> <p>この機能を利用すると、任意のSQLの発行が可能です。</p> <p>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。</p> <p>「管理者」を許可されたユーザ以外は「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」で許可されていないテーブルやビューを利用したSQLを発行できません。</p>
WHERE句の直接入力	<p>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。</p> <p>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。</p> <p>「管理者」を許可されたユーザ以外は「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」で許可されていないテーブルやビューを利用したWHERE句を入力できません。</p>
ルーティングの作成	<p>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。</p> <p>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。</p> <p>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。</p> <p>詳細は「ルーティングの作成」を参照してください。</p>
SQLビルダ	<p>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。</p> <p>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。</p> <p>また、生成AIによるSQL生成を利用する場合においても、「SQLビルダ」が許可されている必要があります。</p> <p>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。</p> <p>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。</p> <p>「管理者」を許可されたユーザ以外は「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」で許可されていないテーブルやビューを利用したSQLを作成できません。</p>



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。

ViewCreator SQLビルダ アシスタント を利用する場合

「ViewCreator SQLビルダ アシスタントによるSQL生成」を利用する場合は、「SQLビルダ」と「ViewCreator SQLビルダ アシスタント」の認可が両方許可されている必要があります。

「ViewCreator SQLビルダ アシスタント」の認可設定を行うには、「リソースの種類」で「IM-Copilot Assistant API」を選択します。

The screenshot shows the 'Access Settings (IM-Copilot Assistant API)' interface. The 'Resource Type' dropdown is highlighted with a red box and set to 'IM-Copilot Assistant API'. Below it is a table with columns for 'Resource', 'Action', 'Authentication', 'Organization', and 'Role'. Two rows are visible: 'IM-Copilot Assistant API' and 'ViewCreator SQLビルダ アシスタント'. Both rows have the 'Execute' action selected. The 'IM-Copilot Assistant API' row has a green checkmark in the 'Role' column for 'IM-Copilot Assistant API Manager'. The 'ViewCreator SQLビルダ アシスタント' row has a green checkmark in the 'Role' column for 'Menu Manager'.

アクセス種別-認可 を利用する場合

データ参照・編集画面で「認可の設定」を利用する場合は、上記の「データ参照の閲覧・作成」「クエリの作成」以外に以下の認可が許可されている必要があります。

- 「リソースの種類」で「画面・処理」を選択し、認可リソース「認可設定(ポップアップ)」、「認可設定(Ajax用)」を許可します。

The screenshot shows the 'Access Settings (画面・処理)' interface. The 'Resource Type' dropdown is highlighted with a red box and set to '画面・処理'. Below it is a table with columns for 'Resource', 'Action', 'Authentication', 'Organization', and 'Role'. Several rows are visible, including 'Forma', 'Formaアプリ作成', 'Formaアプリ作成管理', 'Formaユーザ', 'WF申請書再利用', '認可', '認可設定(基本画面)', '認可設定(ポップアップ)', '認可設定(Ajax用)', 'カレンダー', and 'カレンダー一覧'. The 'Forma' row has a green checkmark in the 'Role' column for 'Forma Manager'. The '認可設定(ポップアップ)' row has a green checkmark in the 'Role' column for 'Menu Manager'.

- 「リソースの種類」で「データ参照のアクセス権設定」を選択し、認可リソースグループ「データ参照のアクセス権設定」を許可します。

リソースの種類	データ参照のアクセス権設定	アクションの種類	全てのアクション	権限設定を開始する
リソース	アクション	認証	組織	ロール
		ゲストユーザ 認証済みユーザ	サンプル会社 その他会社	テナント管理者 認可管理者 メニュー管理者 アカウント管理者 ロール管理者 カレンダーマネージャー ジョブマネージャー管理者 IM共通マスター管理者 IM共通マスタ運用管理者 ポータル管理者 IM-Workflow管理者
データ参照のアクセス権設定	参照			

i コラム

「データ参照のアクセス権設定」は配下の認可リソースが存在しない場合、「認可設定」画面に表示されません。

「データ参照のアクセス権設定」が表示されていない場合は、「テナント管理者」または「ViewCreator管理者」ロールを割り当てられたユーザでアクセス種別-認可のデータ参照を作成してから「認可設定」画面を表示してください。

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。

メニュー設定

エクスポート

グローバルナビ(PC用)

検索キーワードを入力してください。

検索 クリア

メニューの権限設定

- Top
- テナント管理
- ViewCreator
 - クエリ一覧
 - データ参照一覧
 - インポート/エクスポート
 - テーブル・キャプション登録
- サンプル

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

クエリの作成

接続するデータベースを指定し、クエリを新規に作成します。

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

i コラム

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

Flash版のクエリ編集画面は 2024 Autumn(Jasmine) から廃止されました。

今後は、HTML版のクエリ編集画面をご利用ください。

i コラム

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

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

目次

- クエリ一覧画面
 - 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入れ、「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入れ、「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 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。

エクスポートについて、2024 Autumn(Jasmine)以降と2024 Spring(Iris)以前で XML ファイルのフォーマットが異なります。

2024 Autumn(Jasmine)以降では、どちらのフォーマットのファイルもインポート可能です。



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。

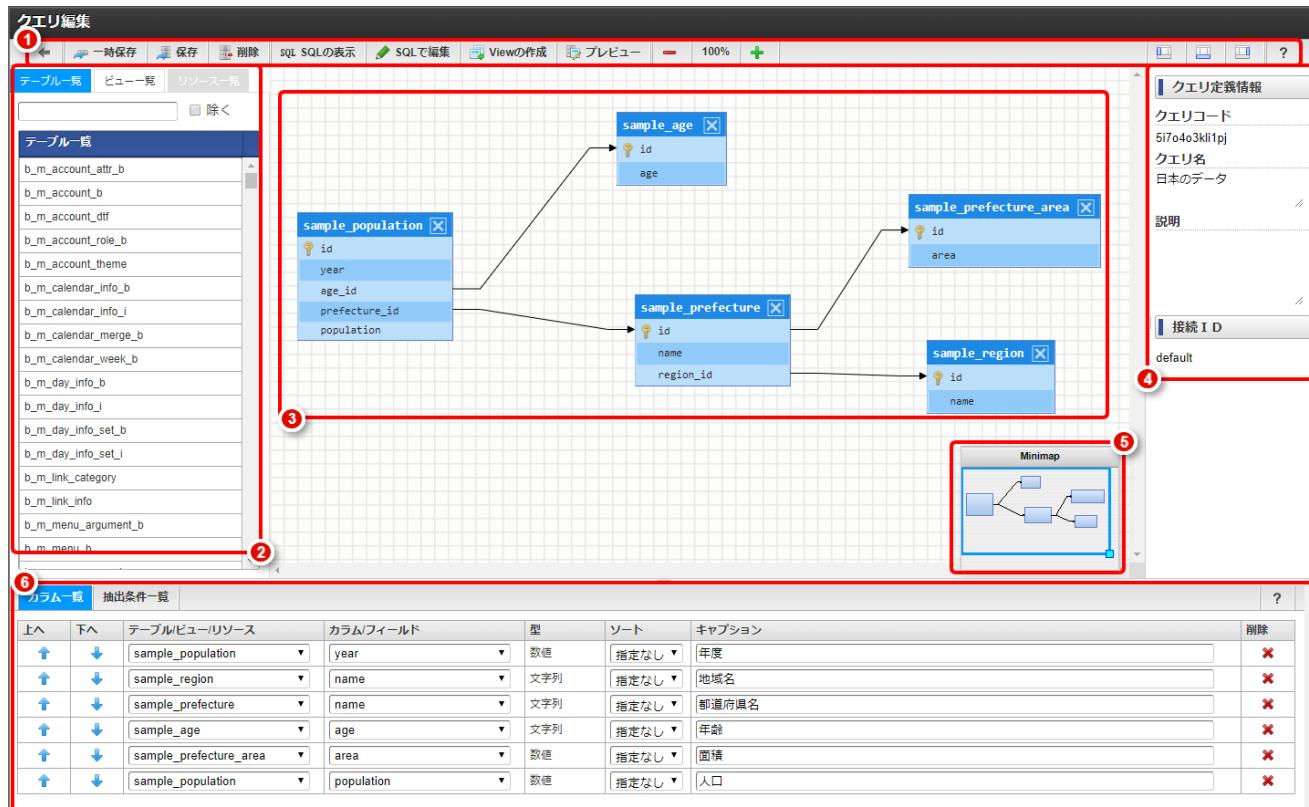
目次

- 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。

項目名	説明
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できます。 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。

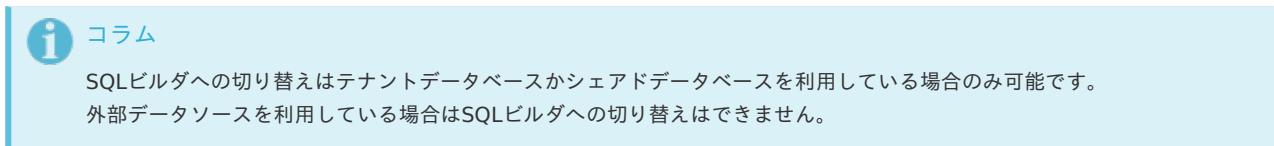
ビューの詳細は「[物理Viewの作成と表示](#)」を参照してください。

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。



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。

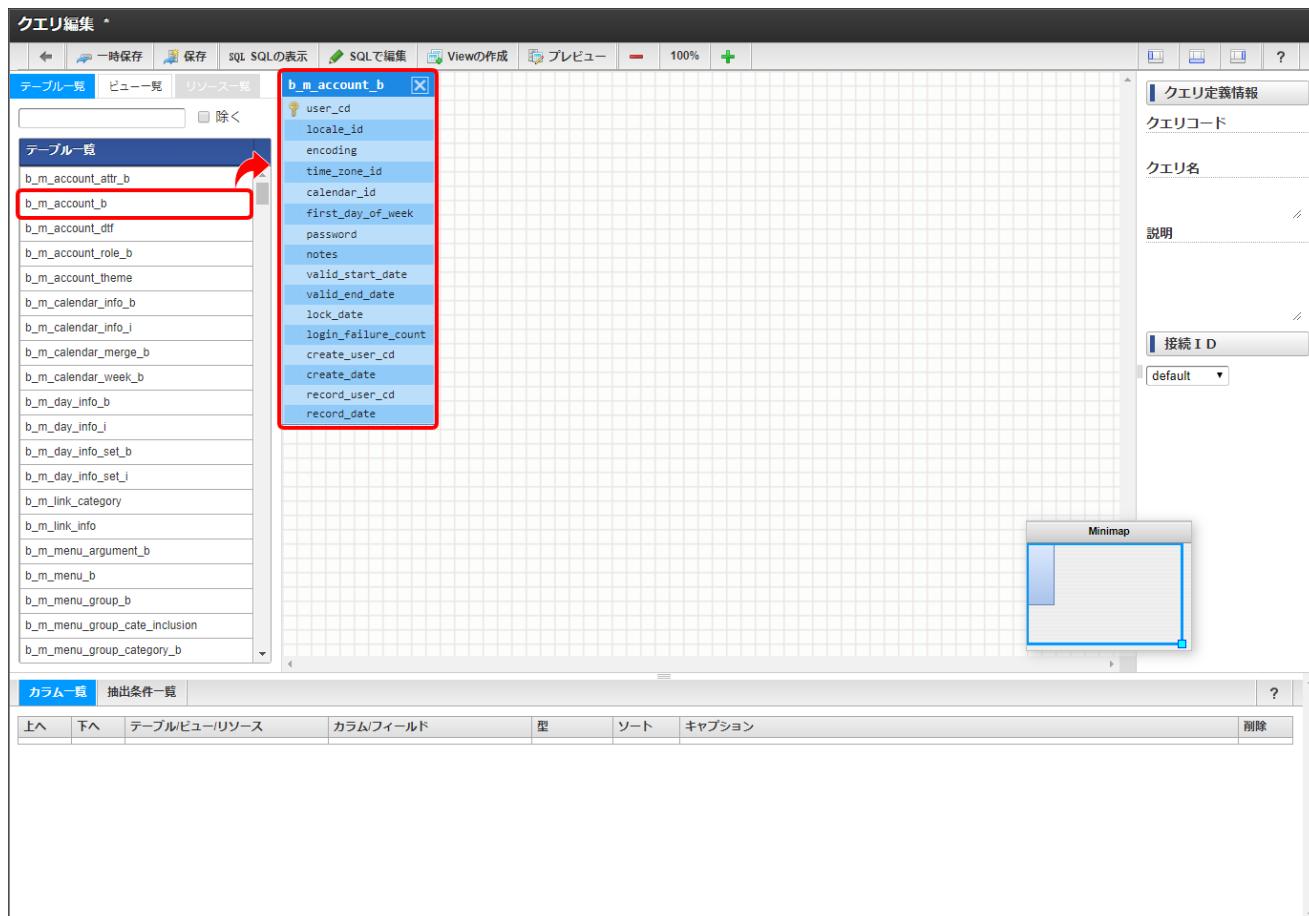
タブを切り替えることでカラム一覧および抽出条件一覧を表示します。

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。



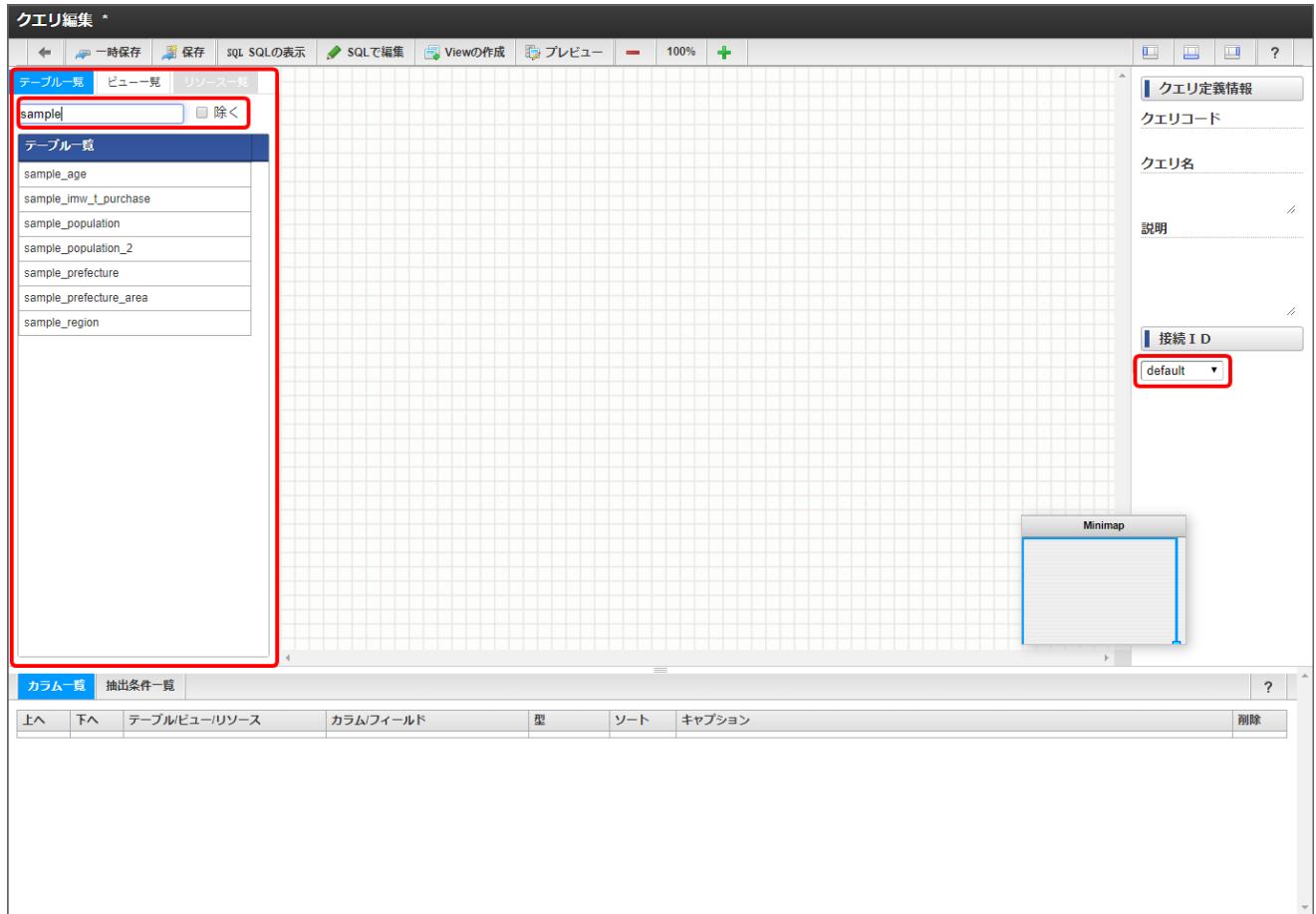
図：クエリ編集画面 - テーブル追加

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

i コラム

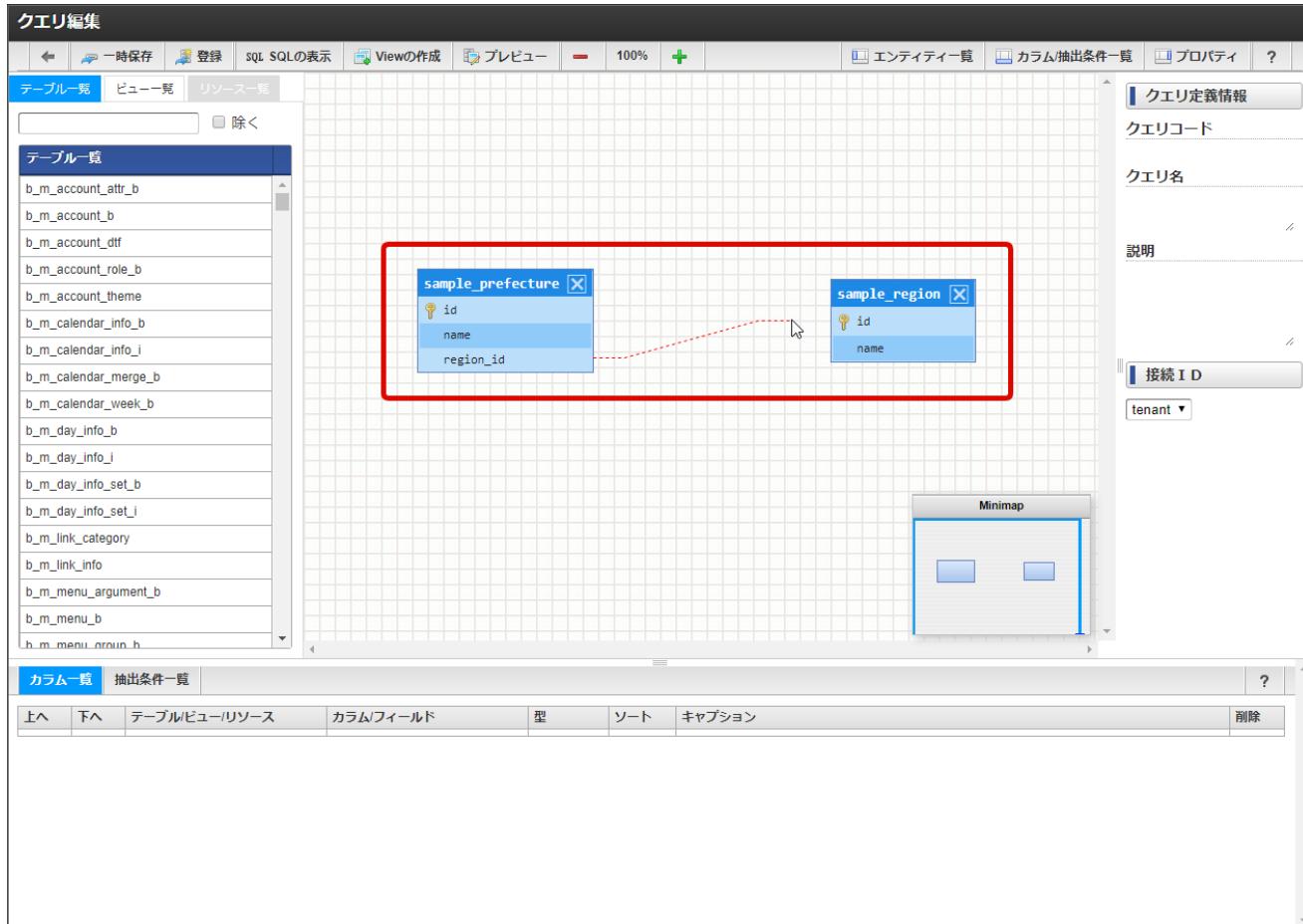
クエリのエクスポートデータ(XML)内の<db-source>タグの内容を書き換えることで接続IDの変更に対応できます。

! 注意

複数の接続IDをまたがったクエリは作成できません。

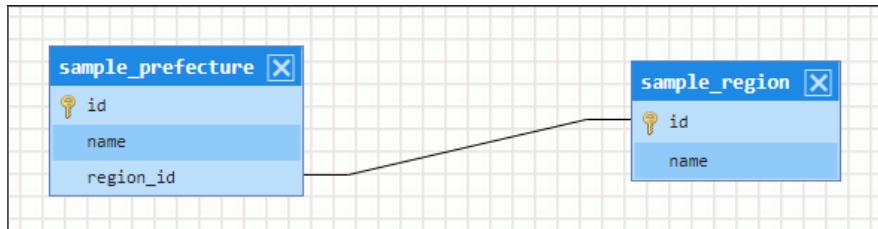
結合条件の作成

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



図：クエリ編集画面 - 結合条件

結合条件（結合線）が描画されます。

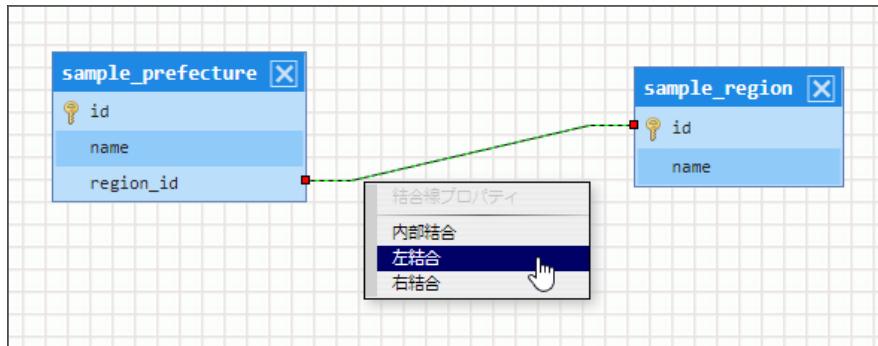


図：クエリ編集画面 - 結合線

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

※初期設定では内部結合(INNER JOIN)です。

また、選択中の結合線はDELETEキーで削除できます。



図：クエリ編集画面 - 結合条件変更

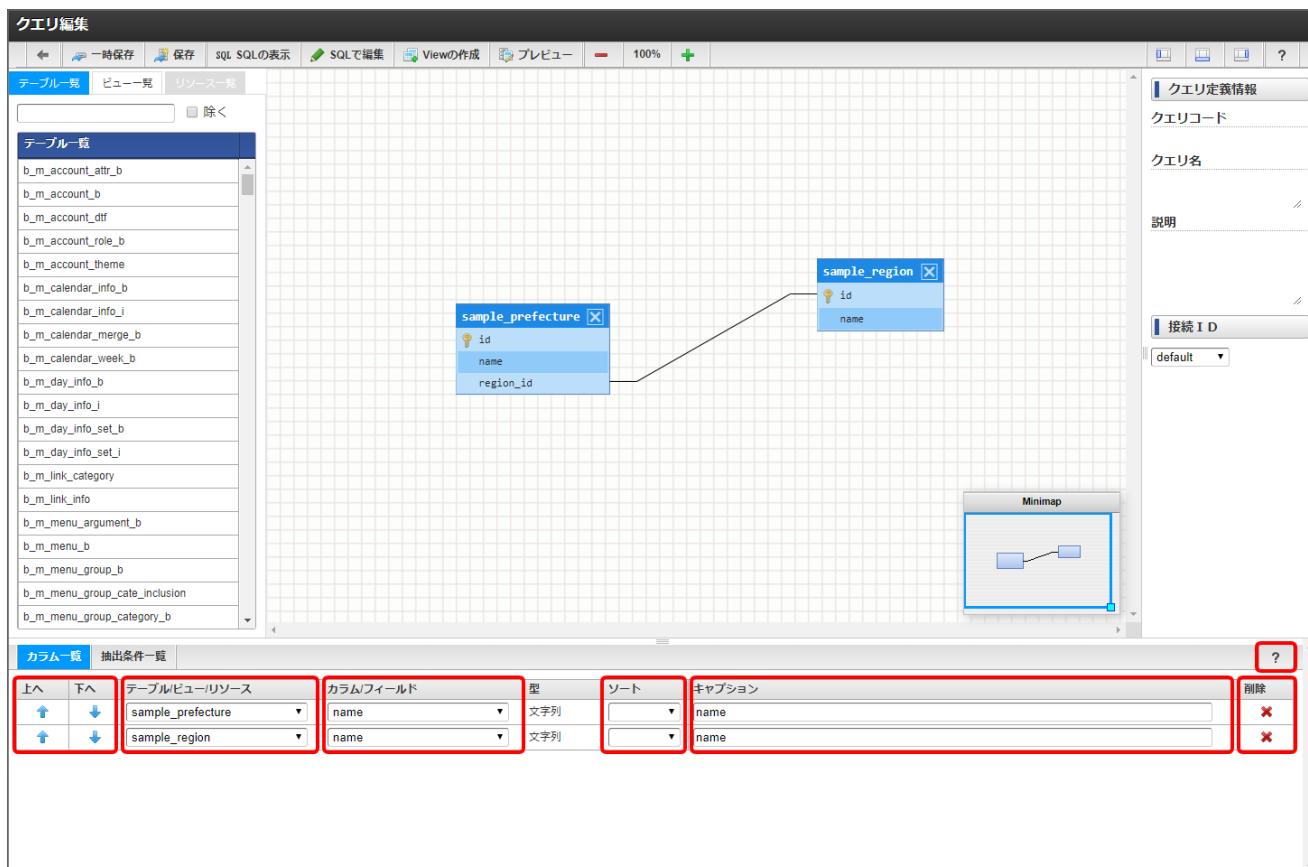


注意

エンティティがリソースの場合は、結合条件を作成できません。

カラムの追加

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

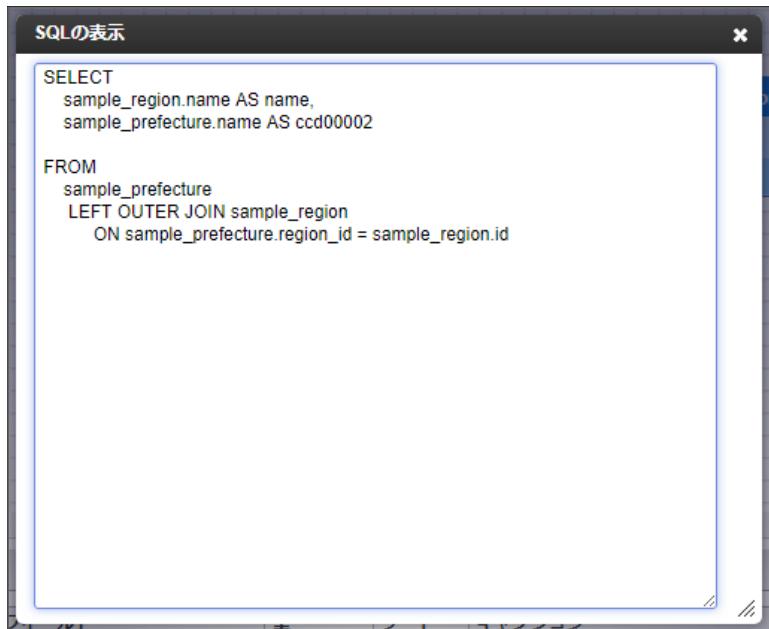


図：クエリ編集画面 - カラム追加

項目名	説明
上へ/下へ	カラムの表示順を入れ替えます。
テーブル/ビュー/リソース	デザイナ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）のサイトツアーゲームが呼び出されます。

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。



```

SELECT
    sample_region.name AS name,
    sample_prefecture.name AS ccd00002

FROM
    sample_prefecture
LEFT OUTER JOIN sample_region
        ON sample_prefecture.region_id = sample_region.id

```

図：クエリ編集画面 - SQLの表示



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

図：クエリ編集画面 - プレビュー



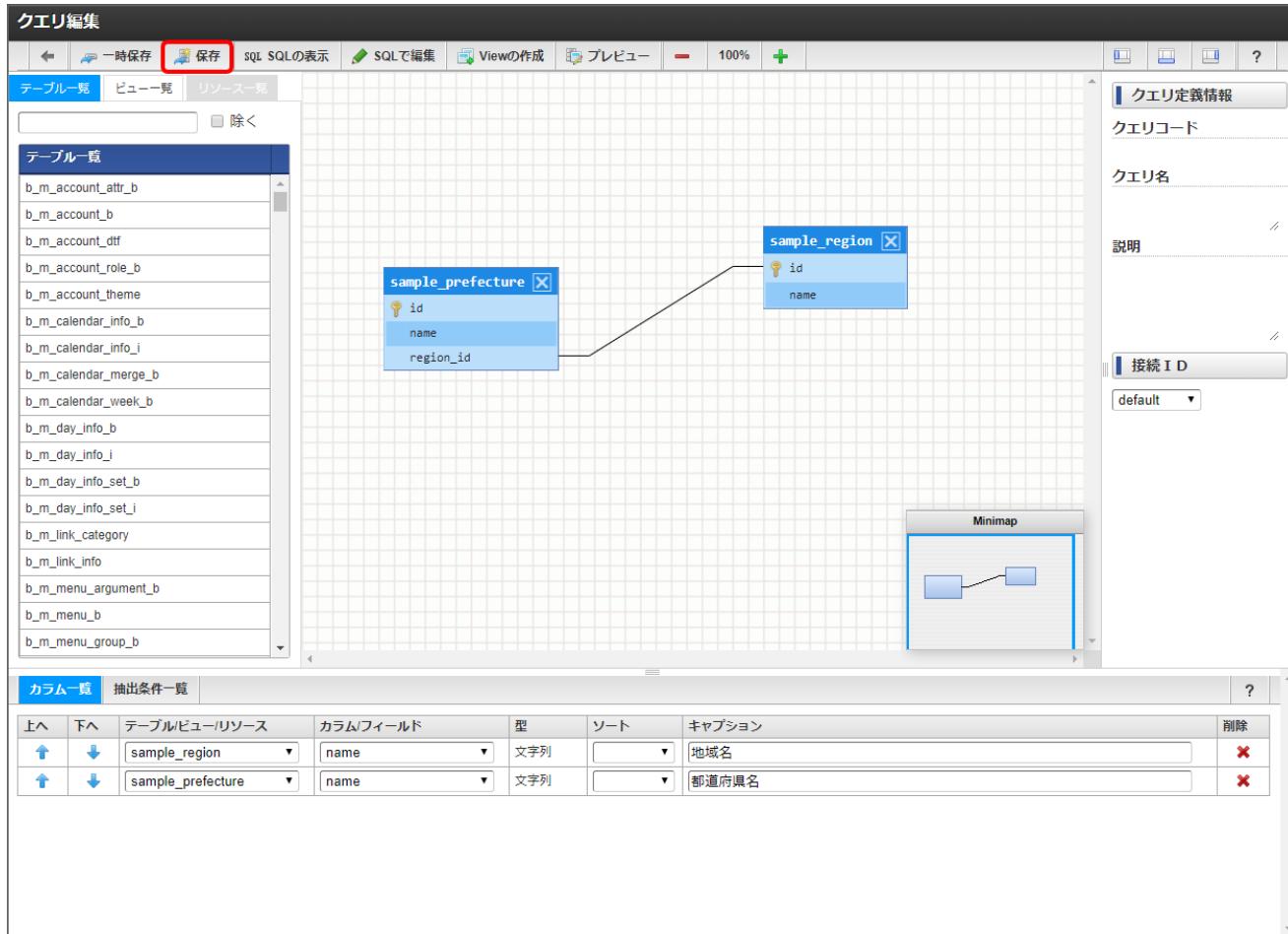
コラム

タイムスタンプ型のカラムは、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。

クエリ定義情報

クエリコード
8fh16zaswe6qchuh

クエリ名
日本のデータ (データ結合)

説明

接続 I D
default

カラム一覧に反映

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

図：クエリ編集画面 - 初期表示

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できます。
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。

クエリ編集 *

← 一時保存 保存 削除 プレビュー ?

```

1 SELECT |
2   pop.year AS year,
3   region.name AS region_name,
4   prefecture.name AS prefecture_name,
5   age.age AS age,
6   prefecture_area.area AS area,
7   pop.population AS population
8
9 FROM (
10   (
11     select
12       *
13     from
14       sample_population
15     union select
16       *
17     from
18       sample_population_2
19   ) AS pop
20   LEFT OUTER JOIN sample_age AS age
21   ON pop.age_id = age.id
22   LEFT OUTER JOIN sample_prefecture AS prefecture
23   ON pop.prefecture_id = prefecture.id
24   LEFT OUTER JOIN sample_prefecture_area AS prefecture_area
25   ON prefecture.id = prefecture_area.id
26   LEFT OUTER JOIN sample_region AS region
27   ON prefecture.region_id = region.id
28

```

クエリ定義情報
クエリコード
8fh6zaswe6qch
クエリ名
日本のデータ（データ結合）
説明
接続 ID
default

カラム一覧 抽出条件一覧

カラムフィールド 型 キャプション

year	数値	年度
region_name	文字列	地域名
prefecture_name	文字列	都道府県名
age	文字列	年齢
area	数値	面積

▲ カラム一覧に反映

⚠️ SQLをカラム一覧に反映してください。

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。

プレビュー

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



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。

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

図：クエリ編集画面 - 保存

クエリ定義情報の保存または更新ダイアログが表示されます。



図：クエリ編集画面 - 保存ダイアログ

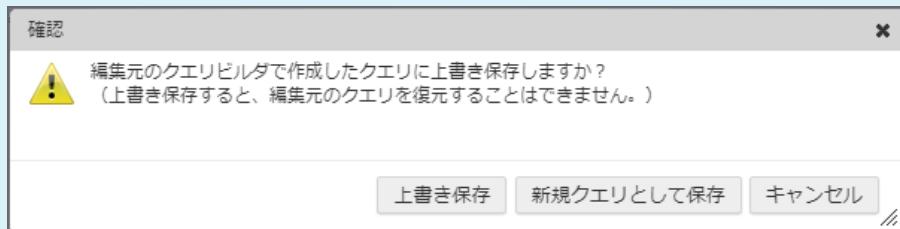
項目名

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。

i 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
- ページ設定
- 権限設定
 - 参照権の設定
 - 認可の設定

データ参照の種類の選択

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

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

データ参照作成	クエリ名	クエリコード	接続 ID	更新日	作成者	エクスポート
	system_log	5ib8yonru57ws	files	6月 12, 2014 10:58 午	system	
	transition_log	5ib8onyuhzy	files	6月 12, 2014 10:58 午	system	
	ViewCreatorの設定テーブルを利用	5i7ur3opljh2x	default	11月 6, 2013 2:40 午	tenant	
	SQL ViewCreatorの設定テーブルを利用	5i7ur3opljh2xa	default	4月 27, 2021 3:45 午	imart	
	日本のデータ	5i7o4o3kli1pj	default	10月 29, 2013 2:26 午	tenant	
	SQL 日本のデータ（データ結合）	8fh16zaswe6qchu	default	2月 6, 2020 3:28 午後	tenant	

クエリ一覧へ

新規 コピー 削除 データ参照一覧へ

検索 クリア

ファイル追加... 中断 削除 インポート

アイコン 遷移先

-  リスト集計（サマリ集計）のデータ参照・新規作成画面に遷移します。
-  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。

エクスポートについて、2024 Autumn(Jasmine)以降と2024 Spring(Iris)以前で XML ファイルのフォーマットが異なります。

2024 Autumn(Jasmine)以降では、どちらのフォーマットのファイルもインポート可能ですが。

リスト集計の作成

リスト集計作成アイコンをクリックすると次の画面に遷移します。

■ データ参照・編集 リスト/サマリ集計
[!\[\]\(36f0d50103f8480e8637a4dd3e66a5a9_img.jpg\) 削除](#) [!\[\]\(ff7950c0921bcb8d9e71431e10af920f_img.jpg\) プレビュー](#)

データ参照設定

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

表示設定

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

カテゴリ 項目名**説明**

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

タイムゾーン設定

タイムゾーン設定の表示	<input type="checkbox"/> タイムゾーン設定を表示する
画面表示のデフォルト・タイムゾーン	(デフォルト)ユーザが利用するタイムゾーン
CSV出力のデフォルト・タイムゾーン	(デフォルト)(GMT+00:00) UTC

カラム一覧

計算式を追加	カラムの国際化項目の編集
カラム	タイプ
地域名(name)	グループ化
都道府県名(name)	グループ化
年度(year)	グループ化
年齢(age)	
面積(area)	平均
人口(population)	合計
キャプション 人口密度	# ## 3桁区切り
式: 人口/面積	fx

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。

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

*ソート順は非表示カラムにも適用させることができます。

- 表示例

地域名	都道府県名	年齢	面積	人口
関東	栃木	0 - 14歳	6,408	339
関東	栃木	65歳以上	6,408	239
関東	栃木	15 - 64歳	6,408	1,352
関東	栃木	0 - 14歳	6,408	307
関東	栃木	15 - 64歳	6,408	1,143
関東	栃木	65歳以上	6,408	167
関東	栃木	0 - 14歳	6,408	380
関東	栃木	65歳以上	6,408	196
関東	栃木	0 - 14歳	6,408	389
関東	栃木	15 - 64歳	6,408	1,070
関東	栃木	15 - 64歳	6,408	1,191
関東	栃木	65歳以上	6,408	141
関東	栃木	0 - 14歳	6,408	414

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できます。

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。



注意

- 表示するデータ参照に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。

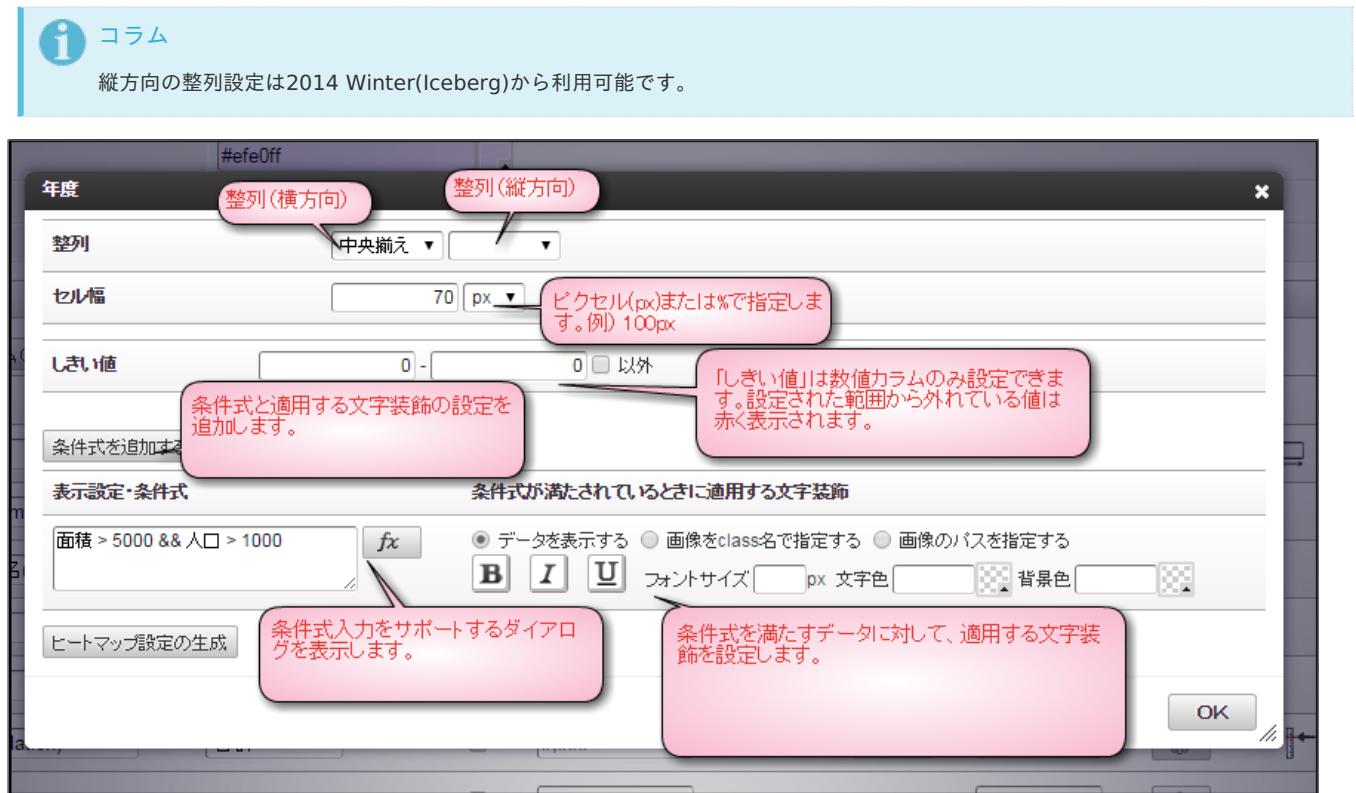
- ## ■ プログラム

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。



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。

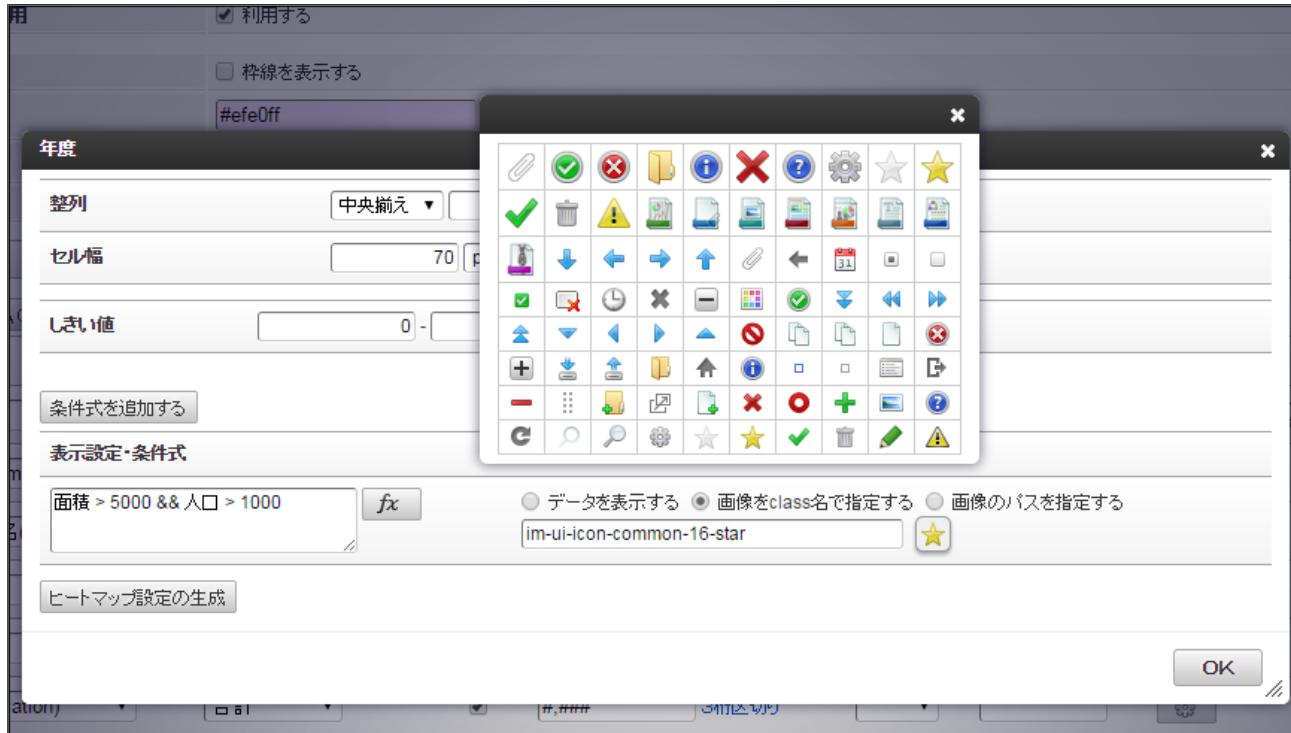


注意

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。

← リストサンプル

年度	地域名	都道府県名	年齢	面積	人口
1970	北海道	北海道	0 - 14歳	83455	1309
1970	東北	青森	0 - 14歳	9235	397
1970	東北	岩手	0 - 14歳	15279	362
1970	東北	宮城	0 - 14歳	6862	449
1970	東北	秋田	0 - 14歳	11434	303
1970	東北	山形	0 - 14歳	7394	288
1970	東北	福島	0 - 14歳	13783	508
1970	関東	茨城	0 - 14歳	6096	534
1970	関東	栃木	0 - 14歳	6408	389
1970	関東	群馬	0 - 14歳	6363	397
1970	関東	埼玉	0 - 14歳	3767	1003
1970	関東	千葉	0 - 14歳	4996	846
1970	関東	東京	0 - 14歳	2102	2401
1970	関東	神奈川	0 - 14歳	2416	1302
1970	中部	新潟	0 - 14歳	10939	573
1970	中部	富山	0 - 14歳	2802	230
1970	中部	石川	0 - 14歳	4185	235
1970	中部	福井	0 - 14歳	4189	178
1970	中部	山梨	0 - 14歳	4201	188
1970	中部	長野	0 - 14歳	12598	450
1970	中部	岐阜	0 - 14歳	10209	427
1970	中部	静岡	0 - 14歳	7329	765
1970	中部	愛知	0 - 14歳	5124	1310

< 1 2 3 4 5 6 7 8 9 10 11 ... 20 > 1 - 50 / 987

プレビューを閉じて、下記のように設定します。

1. 「グループ化」を設定したカラムは、ドリルダウンが可能です。※サマリ集計の場合は動きが変わります
2. 「集計関数」を設定すると、小計や合計が表示されます。
3. 「表示」のチェックを外したカラムは表示されません。
4. 数値型や日付型のカラムは表示フォーマットを設定できます。
5. 「ソート順」を指定すると、データの並び順を制御できます。※ソート順を指定しないと並び順が保証されません
6. 「表示設定」の「整列」で「右寄せ」にします。
7. 「表示設定」の「しきい値」で、3000 - 8000 を指定します。設定した範囲から外れた値は赤く表示されます。

以下、実行結果です。

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

1. トップ階層へ戻ります。
2. クリックした階層へ戻ります。
3. +アイコンをクリックするとグループ化した列が展開されます。
4. 集計関数を設定したカラムについて、小計や合計が表示されます。また、しきい値の範囲外の値が赤く表示されます。



コラム

表全体に適用できる設定項目もいくつかあります。

表示設定	
枠線	<input type="checkbox"/> 枠線を表示する ②
1行おきの背景色	#56aaff ②
表全体の幅	600 px ③
ヘッダ行の改行	<input type="checkbox"/> 改行する

- 「枠線を表示する」は、デフォルトでチェックされています。チェックを外すと非表示にできます。
- 1行おきに行単位で背景色を指定できます。
- 「表全体の幅」は、デフォルト（0）では100%が適用されます。列ごとに指定した「セル幅」通りの表示にならない場合は、「表全体の幅」も併せて設定してください。

実行結果は次のように表示されます。

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

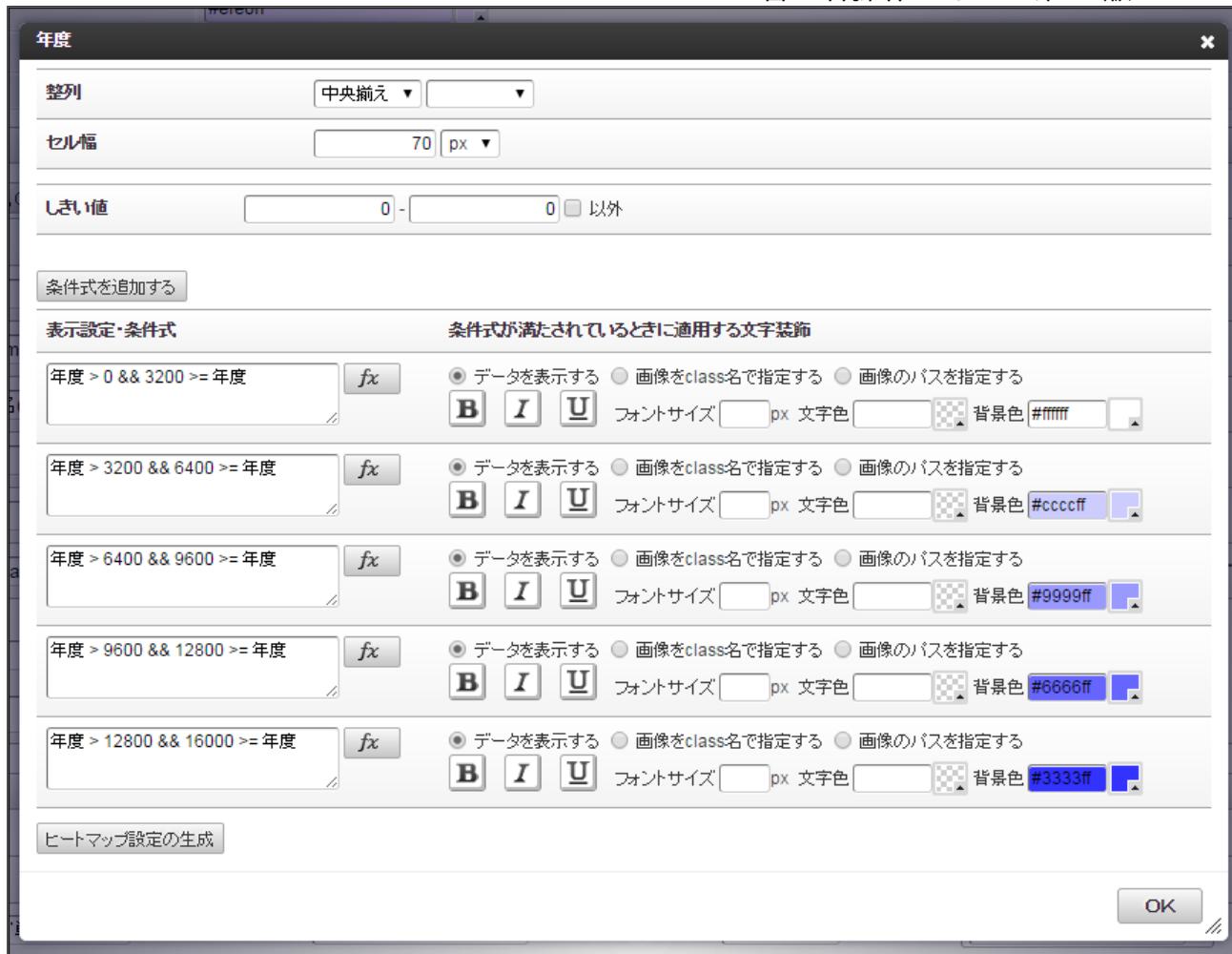
複数の表示条件設定（ヒートマップの作成）

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



コラム

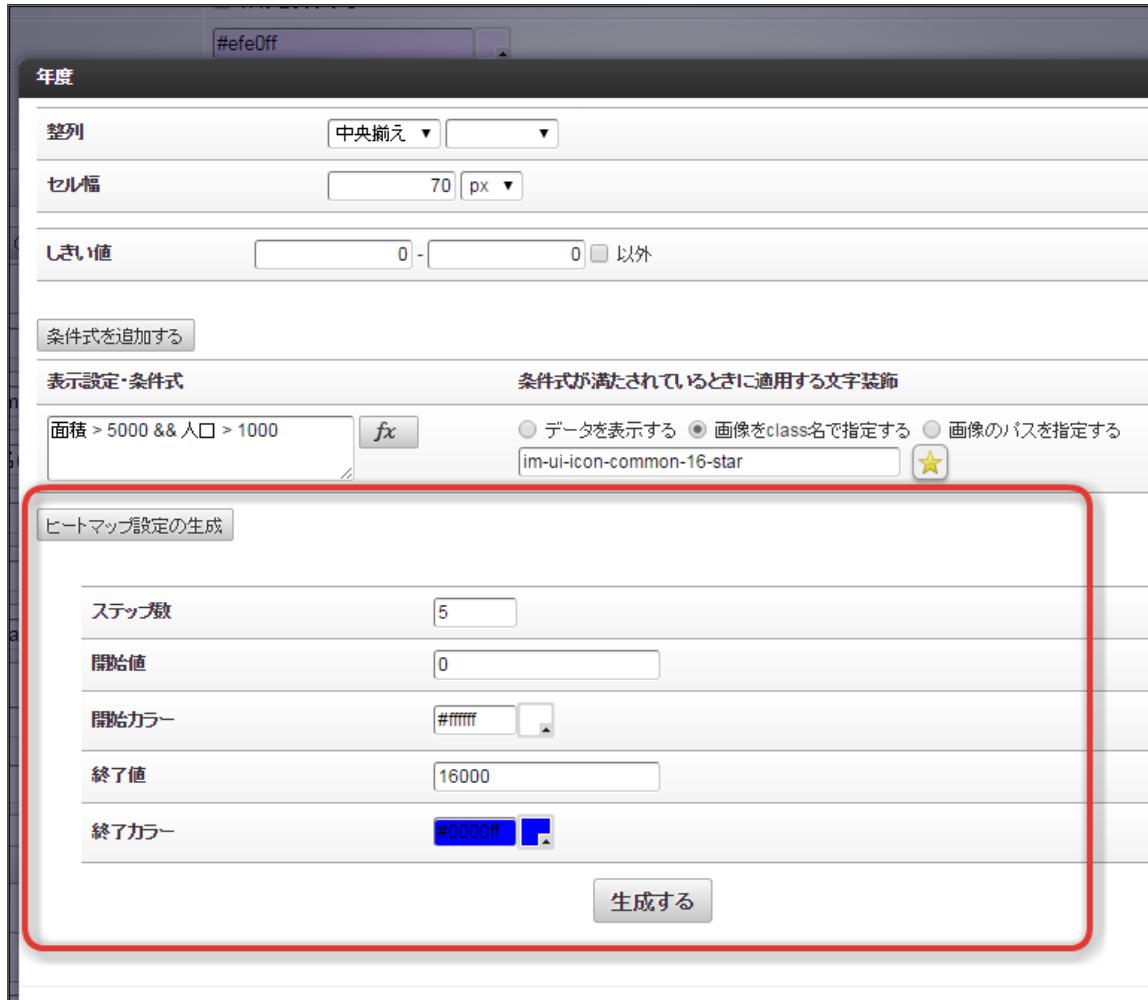
複数の表示条件設定は2014 Summer(Honoka)から利用可能です。



i 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。

クエリ名	日本のデータ
集計パターン	<input style="border: 1px solid #ccc; padding: 2px; margin-right: 10px;" type="button" value="リスト"/> <input style="border: 1px solid #000; background-color: #000; color: white; padding: 2px;" type="button" value="リスト"/> <input style="border: 1px solid #ccc; padding: 2px;" type="button" value="サマリ"/>
説明	

サマリ集計が選択された場合は、グループ化カラムをドリルダウン形式で1階層ずつ展開するのではなく設定されたグループ化列をまとめてグルーピングします。
また、抽出条件の設定を行うことで集合関数を使用できます。

リスト集計からグラフ集計を作成

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

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



コラム

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

いずれの場合においてもグラフ集計作成画面で設定可能な項目は同一です。

データ参照一覧に追加されます。

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

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

データ参照名のリンクをクリックすると、データの閲覧ができます。

年度	地域名	都道府県名	面積(平均)	人口(合計)
田 2000	-	-	7,806	126,699
田 1995	-	-	7,806	125,436
田 1990	-	-	7,806	123,286
田 1985	-	-	7,806	121,017
田 1980	-	-	7,806	116,985
田 1975	-	-	7,806	111,896
田 1970	-	-	7,806	104,665
			7,806	829,984

コラム

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

例) Microsoft Excel 2013の場合は、以下の操作でBOMなしUTF-8のテキストファイルを読み込むことができます。

「データ」タブをクリックします。

「外部データの取り込み・テキストファイル」をクリックします。

読み込むcsvファイルを選択します



注意

2016 Spring(Maxima)より、「リスト集計におけるCSV出力の設定」においてBOMを付与できるようになりました。

そのため、出力されたCSVファイルのファイルエンコーディングがUTF-8の場合でも、Microsoft Excelで開く際にダブルクリック等で開ける可能性があります。

i コラム

CSVファイルのヘッダには、データ参照編集の「カラムの国際化項目の編集」で設定されている値が表示されます。
「カラムの国際化項目の編集」については「[国際化とヘッダ・フッタ](#)」を参照してください。

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可能にする集計方法です。

※下記はクロス集計の表示画面です。編集画面については後述します。

i コラム

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

The screenshot shows the configuration screen for the 'Annual - Prefecture Population Cross-Count' view. The interface includes:

- Top Navigation:** '年度・都道府県別人口 - クロス集計' and a search bar.
- Filter Area (Enclosed in a red box):**
 - 入替え可能なカラム一覧:** A list of columns for selection.
 - 列見出し:** A section for column headers, with '年度' selected and 'ソート指定無し' (Sort Unspecified) chosen.
 - 行見出し:** A section for row headers, with '地域名' and '都道府県名' selected, both with '見出し名' (Header Name) and 'ソート指定無し' (Sort Unspecified).
 - Value Selection:** A section for values, with '人口' selected and '合計' (Sum) chosen.
- Bottom Filter Options:** Buttons for '表示/非表示' (Show/Hide), '合計' (Sum), '平均' (Average), '最小' (Min), '最大' (Max), and 'カウント' (Count).
- Advanced Filter Area:** Includes operators (AND/OR), dropdowns for '地域名' (Regions), '人口' (Population), and '年度' (Year), and date range inputs for year ranges.
- Buttons:** '表示' (Display) and 'クリア' (Clear).
- Data Table:** A grid showing population data for various regions over time. The table has columns for years (1975, 1980, 1985, 1990, 1995, 2000) and summary rows (合計, 平均). The data includes:

		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”的レコードは3つあります。

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

⇒ 合計、平均、最大、最小、カウント、標準偏差、分散

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

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

小計行の表示

「行見出し」に複数のカラムが設定されている場合（カテゴリ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（カテゴリ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。

	1975	1980	1985	1990	1995	2000	合計	平均
北海道	5,338	5,574	5,678	5,634	5,687	5,657	33,568	1,864.89
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

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。

The screenshot shows the configuration interface for a cross-summary report. At the top, there are sections for 'Replaceable Column Selection' (with 'Area' selected), 'Row Header' (with 'Year' and 'Age' filters), and 'Column Header' (with 'Population' and 'Sum'). Below these are sections for 'Region' and 'Population' with dropdown menus for selection. Underneath is a section for 'Combine items with AND/OR logic' (set to AND) and checkboxes for various regions. At the bottom are 'Display' and 'Clear' buttons, followed by a large table showing population data for various regions (北海道, 東北, 中部, 近畿, 四国, 九州) across years (1990-2000) and age groups (0-14, 15-64, 65+).

		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3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。



棒グラフ

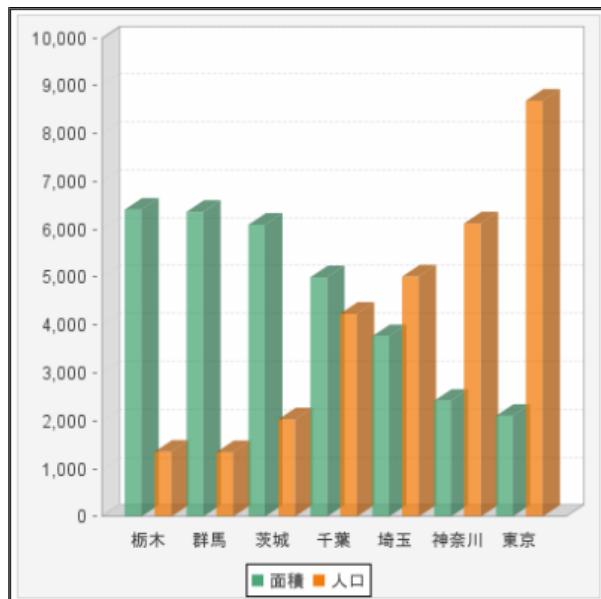


図 棒グラフィイメージ

折れ線グラフ

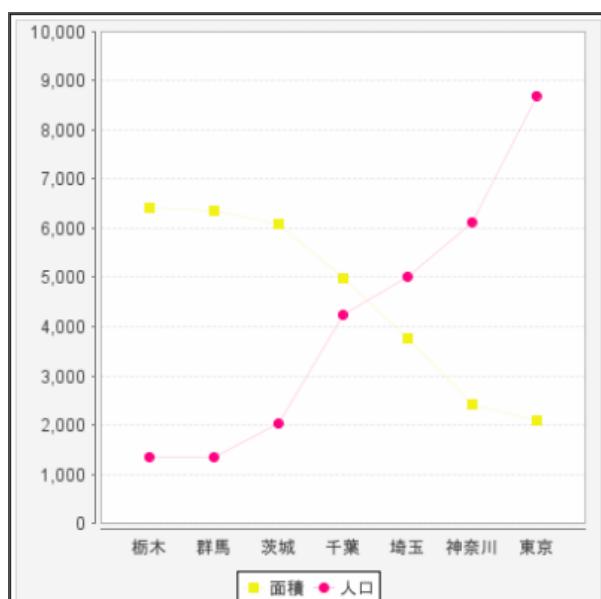


図 折れ線グラフィイメージ

レーダーチャート

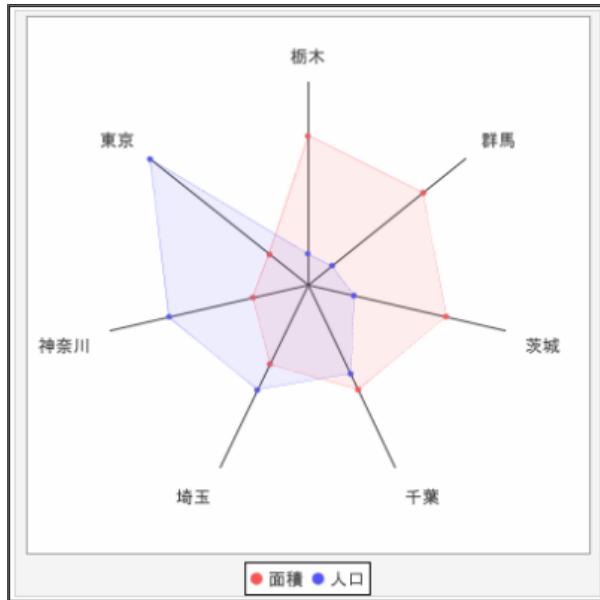


図 レーダーチャートイメージ

円グラフ

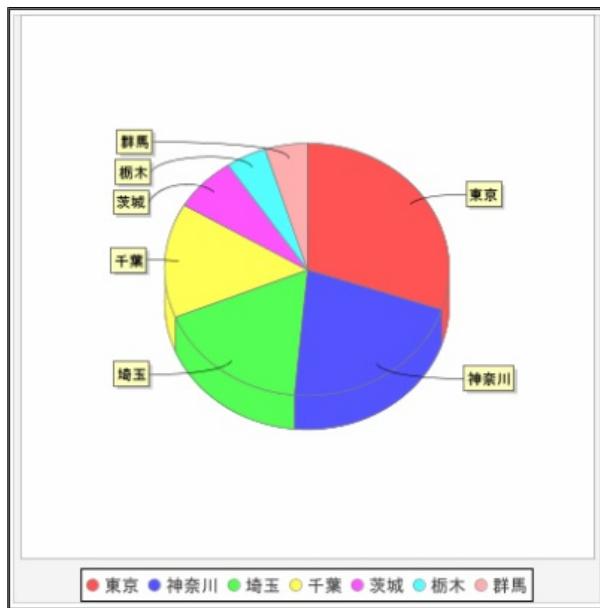


図 円グラフイメージ

グラフ設定

グラフ表示時のレイアウトを設定します。

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

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

i 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。

カラム設定

カラムの国際化項目の編集

キャプションカラム	地域名(name)
凡例コードカラム	都道府県名(name)
凡例ラベルカラム	都道府県名(name)
値カラム	面積(area)

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

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。
カラム	グラフを表示した場合の実データとして利用するカラムです。 円グラフの場合、表示されません。
カラー	グラフ描画する際の色彩を設定します。 円グラフの場合、表示されません。

i コラム

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

Highcharts

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

Highchartsでは5種類のグラフを組み合わせて作成できます。

i コラム

URL (2019年6月現在)

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

グラフ

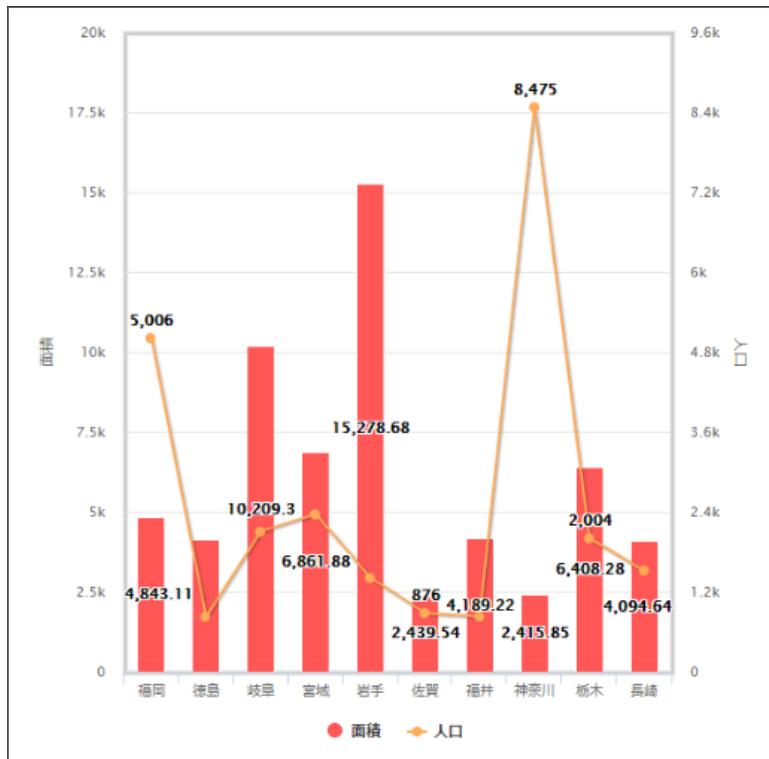


図 グラフ組み合わせイメージ

棒グラフ

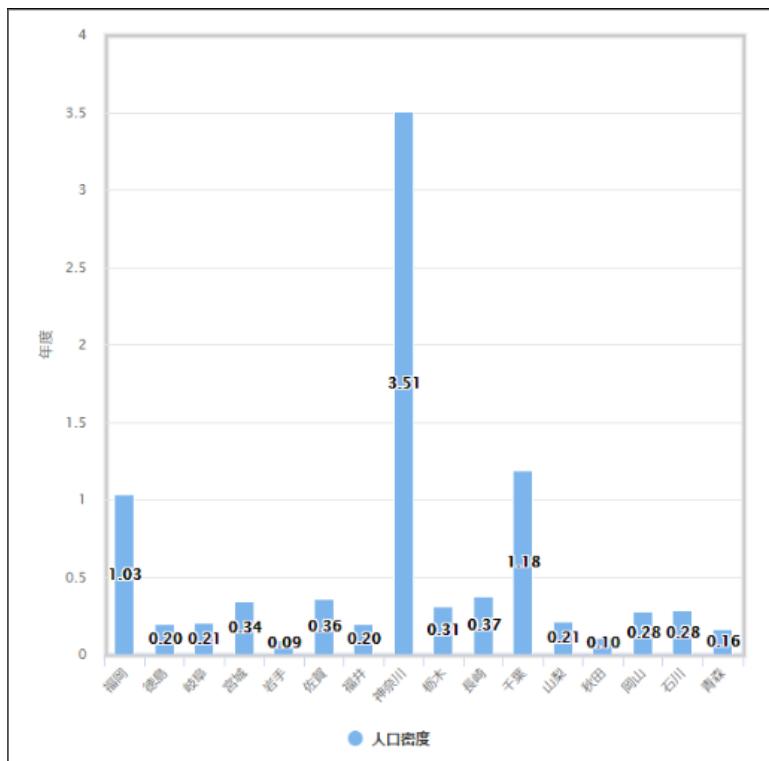


図 棒グラフィイメージ

折れ線グラフ

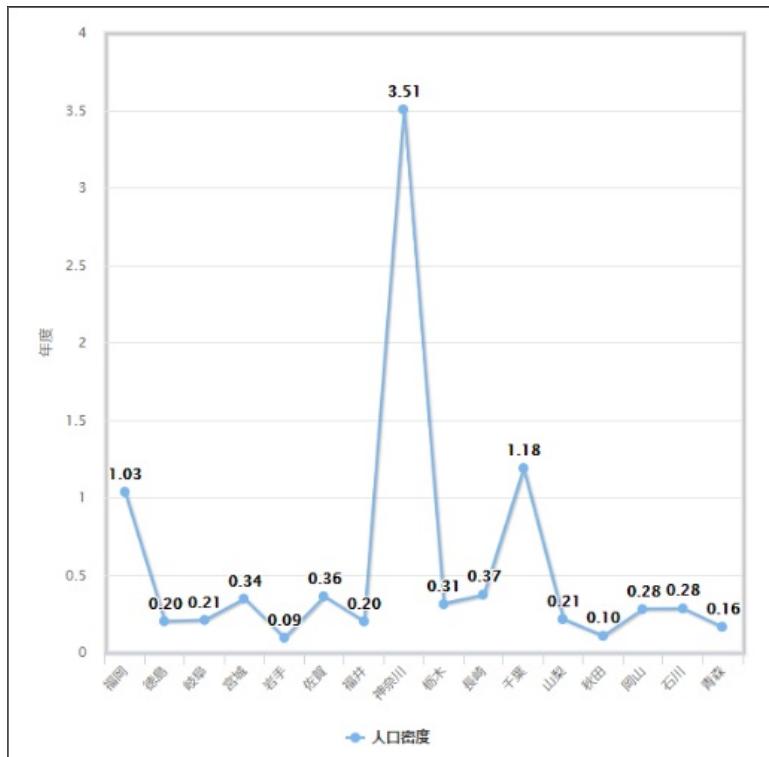


図 折れ線グラフイメージ

レーダーチャート

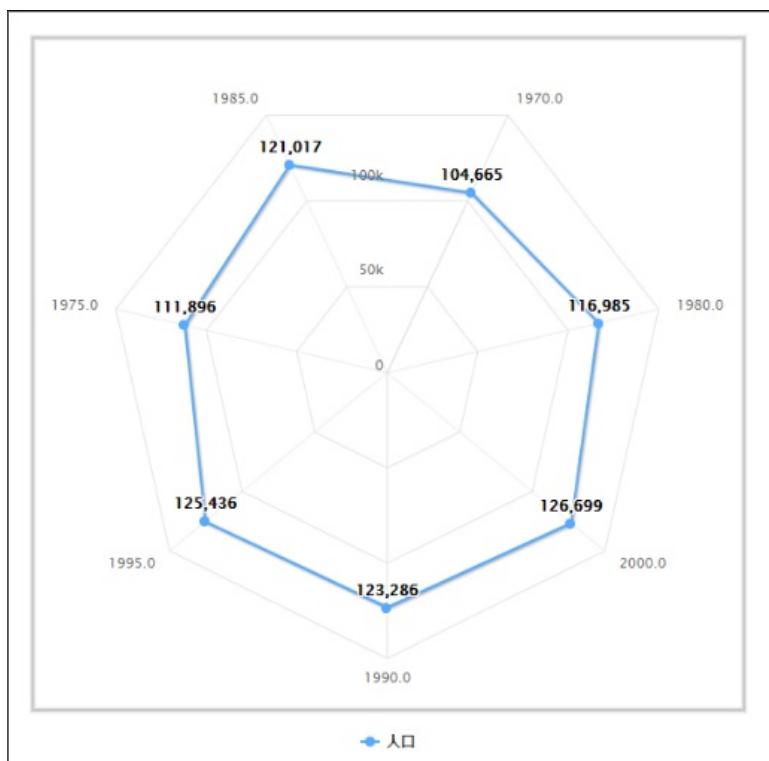


図 レーダーチャートイメージ

円グラフ

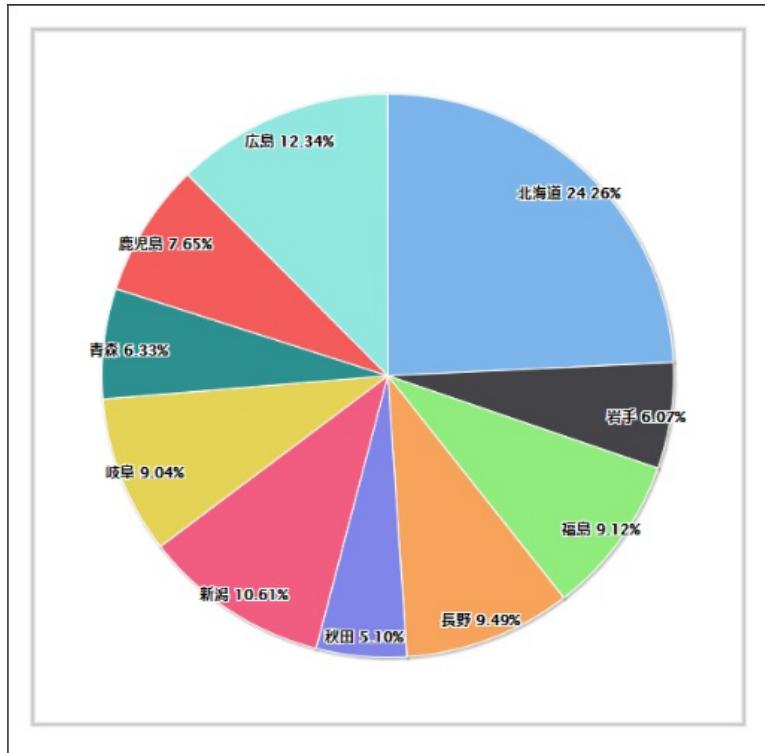


図 円グラフィイメージ

エリアチャート

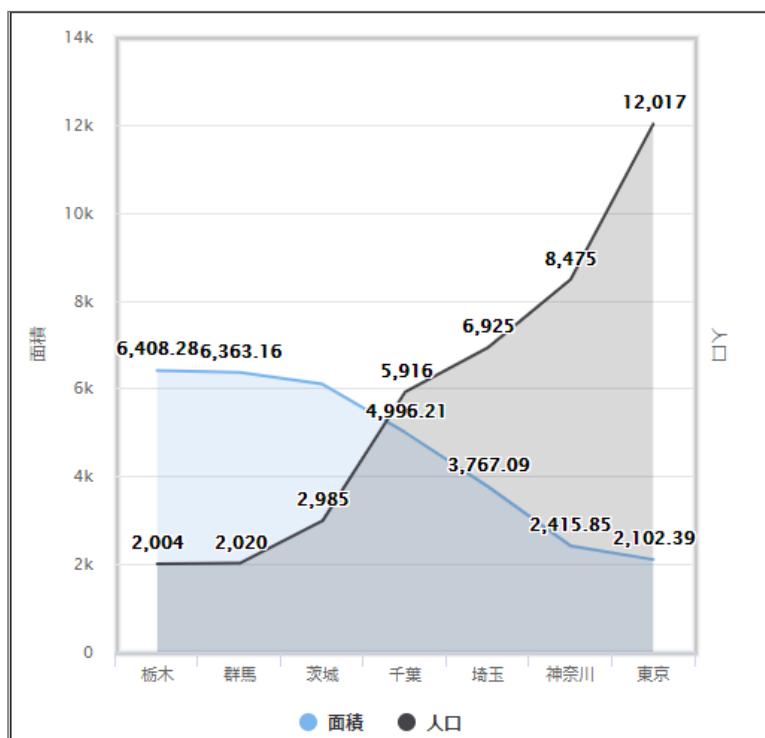


図 エリアチャートイメージ

グラフ設定

グラフ表示時のレイアウトを設定します。

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

図 データ参照・編集画面 グラフ設定項目

設定項目	
項目名	説明
グラフ描画形式	グラフの描画形式を選択します。 intra-mart Accel Platform標準では「 JFreeChart 」、「 Highcharts 」、「 ECharts 」3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できます。

カラム設定

グラフデータとして使用するデータが格納されているカラムを選択します。
クエリに追加されたカラムの内、数値型のカラムのみが表示されます。
データ毎に表示するグラフの種類を選択できます。

カラム設定					
カラムの国際化項目の編集					
キャプションカラム	地域名(name) ▼				
キャプション軸	<input checked="" type="radio"/> 横軸 <input type="radio"/> 縦軸				
凡例コードカラム	都道府県名(name) ▼				
凡例ラベルカラム	都道府県名(name) ▼				
表示	カラム	グラフタイプ	積み上げ表示	目盛り表示位置	カラー
	<input checked="" type="checkbox"/>	年度	棒グラフ ▼	グループ1 ▼	左側 ▼ #ffd4aa
	<input checked="" type="checkbox"/>	面積	折れ線グラフ ▼	▼	右側 ▼ #aad4ff
	<input checked="" type="checkbox"/>	人口	円グラフ ▼	▼	
	<input type="checkbox"/>	人口密度	レーダーチャート ▼	▼	#aaffd4

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

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。

i 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。

ECharts

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

EChartsでは5種類のグラフを組み合わせて作成できます。

i コラム

URL (2024年6月現在)

ECharts : <https://echarts.apache.org/>

グラフ

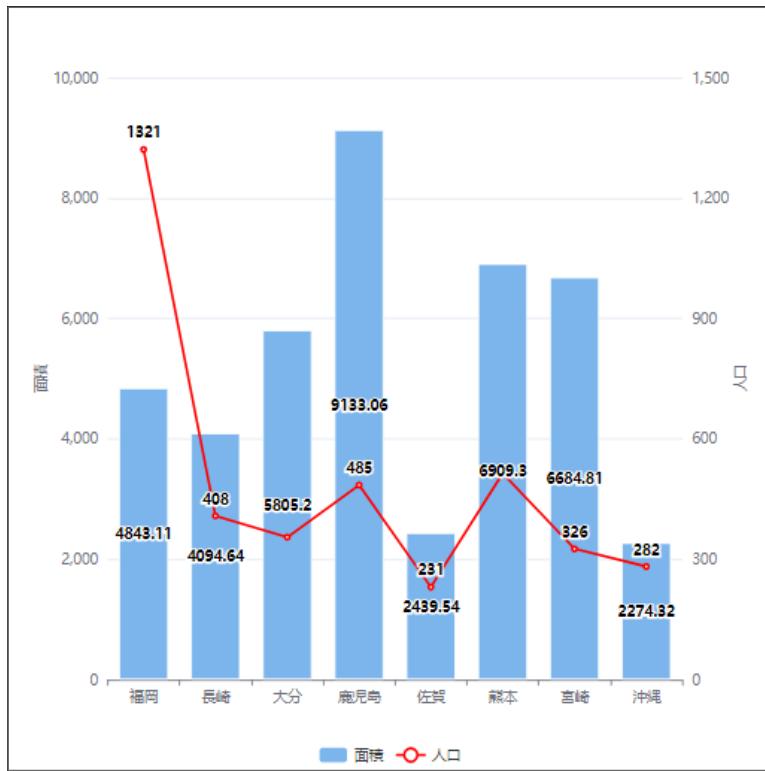


図 グラフ組み合わせイメージ

棒グラフ

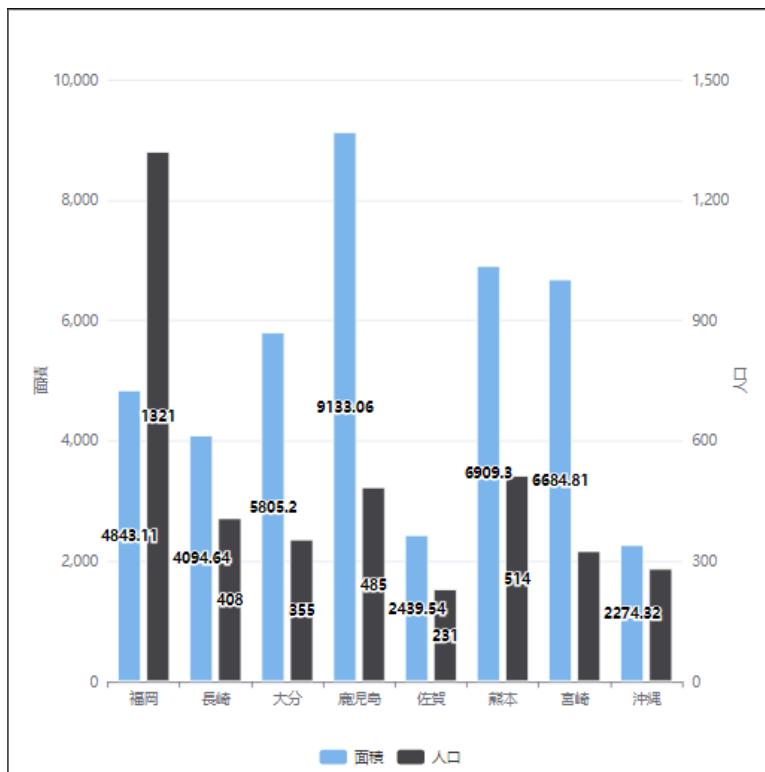


図 棒グラフイメージ

折れ線グラフ

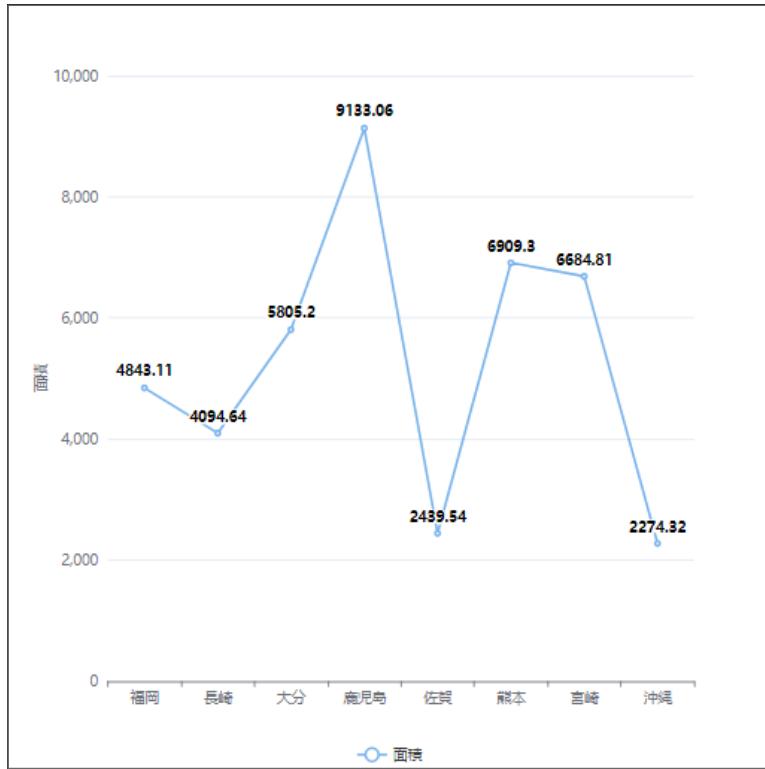


図 折れ線グラフイメージ

レーダーチャート

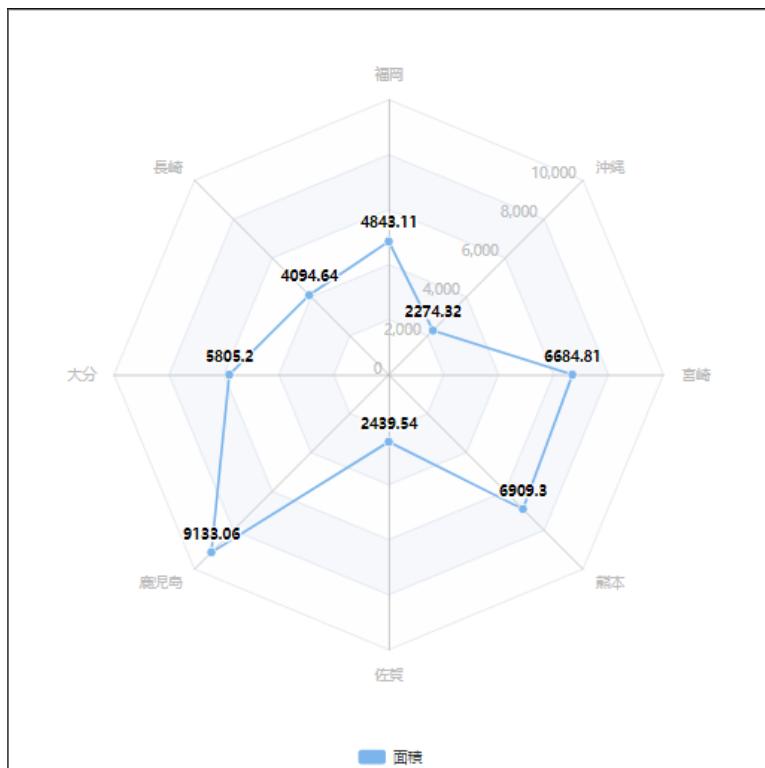


図 レーダーチャートイメージ

円グラフ

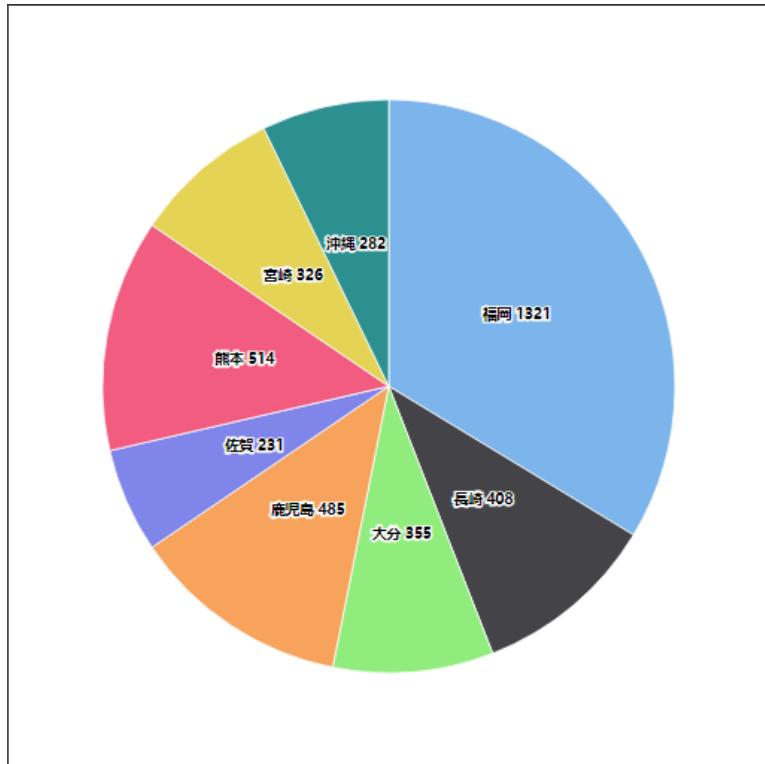


図 円グラフィイメージ

エリアチャート

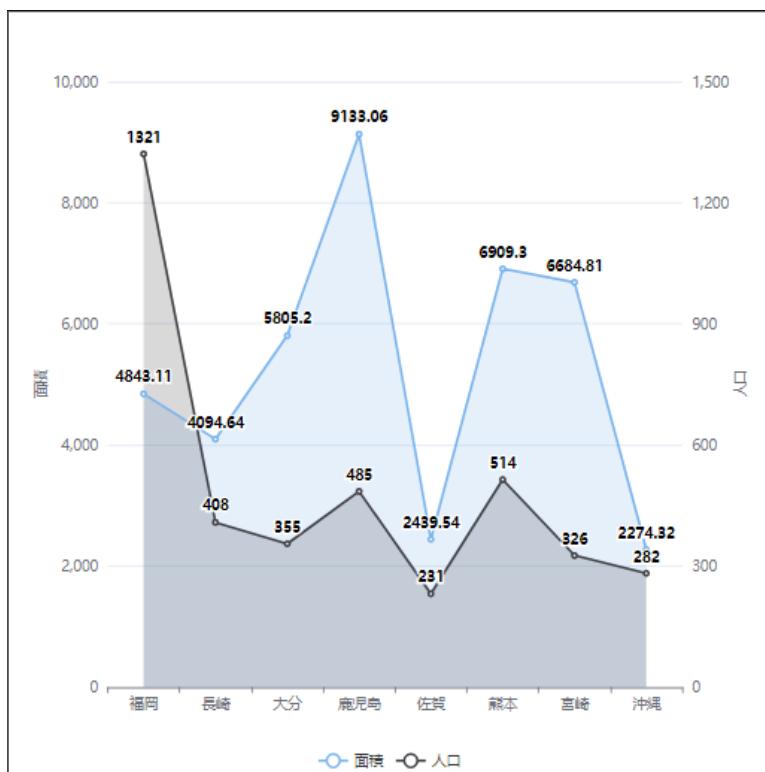


図 エリアチャートイメージ

グラフ設定

グラフ表示時のレイアウトを設定します。

グラフ設定

グラフ描画形式*	ECharts
グラフの大きさ（縦幅）	500
グラフの大きさ（横幅）	500
目盛りの最大値	15000
目盛りの最小値	0
1目盛りあたりの数	3500
100分率表示	<input type="checkbox"/>
リストの表示	<input checked="" type="checkbox"/>

図 データ参照・編集画面 グラフ設定項目

設定項目	
項目名	説明
グラフ描画形式	グラフの描画形式を選択します。 intra-mart Accel Platform標準では「 JFreeChart 」、「 Highcharts 」、「 ECharts 」3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。
目盛りの最大値	グラフ内のデータの中で最も大きい数値を設定します。 目盛りの最大値 = 目盛りの最小値の場合、グラフの上限値と下限値は自動計算されます。
目盛りの最小値	グラフ内のデータの中で最も小さい数値を設定します。 目盛りの最大値 = 目盛りの最小値の場合、グラフの上限値と下限値は自動計算されます。
1目盛りあたりの数	設定された値ごとに数字付きの目盛りが設定されます。 0以下の値が設定された場合は自動計算されます。
100分率表示	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できます。
 円グラフの場合、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。
 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。
 データ毎に表示するグラフの種類を選択できます。

カラム設定

カラムの国際化項目の編集

キャプションカラム	都道府県名(prefecture_name)				
キャプション軸	<input checked="" type="radio"/> 横軸 <input type="radio"/> 縦軸				
凡例コードカラム					
凡例ラベルカラム					
表示	カラム	グラフタイプ	積み上げ表示	目盛り表示位置	カラー
▲▼	✓ 年度	棒グラフ	グループ2	左側	
▲▼	✓ 面積	折れ線グラフ		右側	#00bf5f
▲▼	✓ 人口	エリアチャート	グループ1	右側	#ff0000

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。 Echarts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、レーダーグラフの場合、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。
- 「Echarts」は 2024 Autumn(Jasmine)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。

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ます。

i コラム

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

The screenshot shows the 'Search Settings' dialog box with the following numbered labels:

- 1.**追加ボタン: Adds a new search setting.
- 2.**各項目を結合する演算子: Combines items using AND or OR operators.
- 3.**参照画面の操作状態: Operation status of the reference screen.
- 4.**削除ボタン: Deletes the selected search setting.
- 5.**上へ、下へボタン: Changes the order of search settings by dragging and dropping.
- 6.**検索対象カラム: Column to compare search values.
- 7.**検索タイプ: Search type (e.g., Keyword search).
- 8.**入力制限: Input limit.
- 9.**検索パターン: Search pattern.
- 10.**検索条件デフォルト値: Default value for search conditions.
- 11.**検索条件値・選択肢リスト: Value and selection list for search conditions.

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できます。
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。

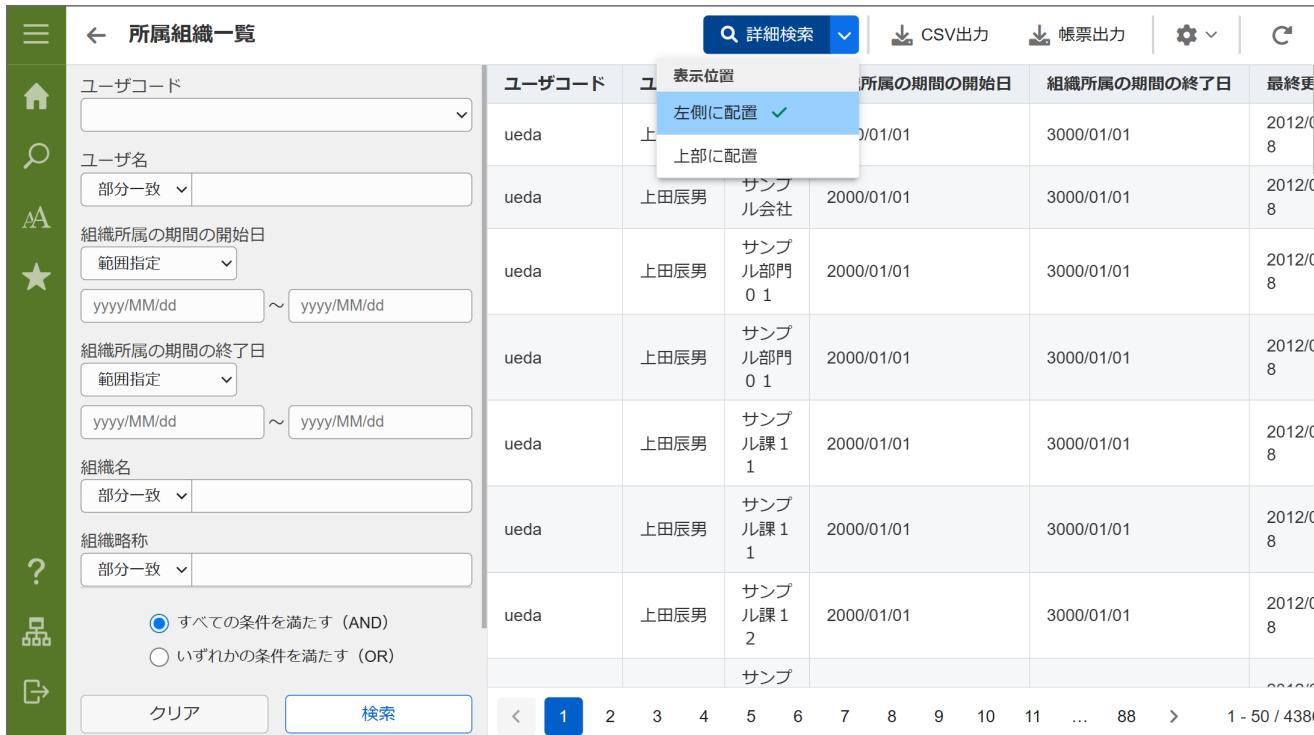
- リスト集計の場合



所属組織一覧									
ユーザコード	ユーザ名	組織名	組織所属の期間の開始日	組織所属の期間の終了日	最終更新日	組織略称	組織検索名	ソートキー	
ueda	上田辰男	サンプル会社	2000/01/01	3000/01/01	2012/08/08	サンプル会社	-	1	
ueda	上田辰男	サンプル会社	2000/01/01	3000/01/01	2012/08/08	サンプル会社	-	1	
ueda	上田辰男	サンプル部門 0 1	2000/01/01	3000/01/01	2012/08/08	サンプル部門 0 1	-	1	
ueda	上田辰男	サンプル部門 0 1	2000/01/01	3000/01/01	2012/08/08	サンプル部門 0 1	-	1	
ueda	上田辰男	サンプル課 1 1	2000/01/01	3000/01/01	2012/08/08	サンプル課 1 1	-	1	
ueda	上田辰男	サンプル課 1 1	2000/01/01	3000/01/01	2012/08/08	サンプル課 1 1	-	1	
ueda	上田辰男	サンプル課 1 2	2000/01/01	3000/01/01	2012/08/08	サンプル課 1 2	-	1	
ueda	上田辰男	サンプル課 1 2	2000/01/01	3000/01/01	2012/08/08	サンプル課 1 2	-	1	
ueda	上田辰男	サンプル部門 0 2	2000/01/01	3000/01/01	2012/08/08	サンプル部門 0 2	-	1	
		サンプル			2012/08/08	サンプル			

リスト集計とサマリ集計では検索欄の位置を選択できます。

「左側に配置」を選択した場合



所属組織一覧									
ユーザコード	表示位置	所属の期間の開始日	組織所属の期間の終了日	最終更新日					
ueda	左側に配置 ✓	2000/01/01	3000/01/01	2012/08/08					
ueda	上部に配置								
ueda	上田辰男	サンプル会社	2000/01/01	3000/01/01	2012/08/08				
ueda	上田辰男	サンプル部門 0 1	2000/01/01	3000/01/01	2012/08/08				
ueda	上田辰男	サンプル部門 0 1	2000/01/01	3000/01/01	2012/08/08				
ueda	上田辰男	サンプル課 1 1	2000/01/01	3000/01/01	2012/08/08				
ueda	上田辰男	サンプル課 1 1	2000/01/01	3000/01/01	2012/08/08				
ueda	上田辰男	サンプル課 1 2	2000/01/01	3000/01/01	2012/08/08				
		サンプ			2012/08/08				

「上部に配置」を選択した場合

← 所属組織一覧

検索条件

- ユーザコード: ユーザーID
- 組織所属の期間の開始日: 年月日
- 組織名: 部門名
- 組織所属の期間の終了日: 年月日
- 組織略称: 略称

検索オプション: 上部に配置 (選択済)

検索ボタン: クリア、検索

ユーザコード	ユーザ名	組織名	組織所属の期間の開始日	組織所属の期間の終了日	最終更新日	組織略称	組織検索名	ソートキー
ueda	上田辰男	サンプル会社	2000/01/01	3000/01/01	2012/08/08	サンプル会社	-	1
ueda	上田辰男	サンプル会社	2000/01/01	3000/01/01	2012/08/08	サンプル会社	-	1
ueda	上田辰男	サンプル部門01	2000/01/01	3000/01/01	2012/08/08	サンプル部門01	-	1
ueda	上田辰男	サンプル部門01	2000/01/01	3000/01/01	2012/08/08	サンプル部門01	-	1
ueda	ト中辰里	サンプル	2000/01/01	3000/01/01	2012/08/08	サンプル	-	1

ページナビゲーション: < 1 2 3 4 5 6 7 8 9 10 11 ... 88 > 1 - 50 / 4386

■ クロス集計の場合

年度・都道府県別人口 - クロス集計

検索、帳票出力

検索条件

- 入れ替え可能カラム一覧: 面積 (列見出し: 年度、年齢)
- 行見出し: 地域名 (見出し名: リスト)、都道府県名 (見出し名: リスト)
- 値: 人口 (合計: 数値フォーマット)

表示/非表示: 合計 (選択済)、平均、最小、最大、カウント

各項目を結合する演算子: AND (選択済)、OR

地域名: 北海道、東北、関東、中部、近畿、中国、四国、九州

人口: 1990 - 2000

年度: 1990 - 2000

表示、クリア

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



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	利用可能な型
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。	文字列
	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。	数値、日付、タイムスタンプ
	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	文字列



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。

ユーザ名
部分一致

ユーザの有効期間の開始日
範囲指定

yyyy/MM/dd ~ yyyy/MM/dd

3	2025					
日	月	火	水	木	金	土
1	/dd					
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

(すべての条件を満たす (AND))
(いずれかの条件を満たす (OR))

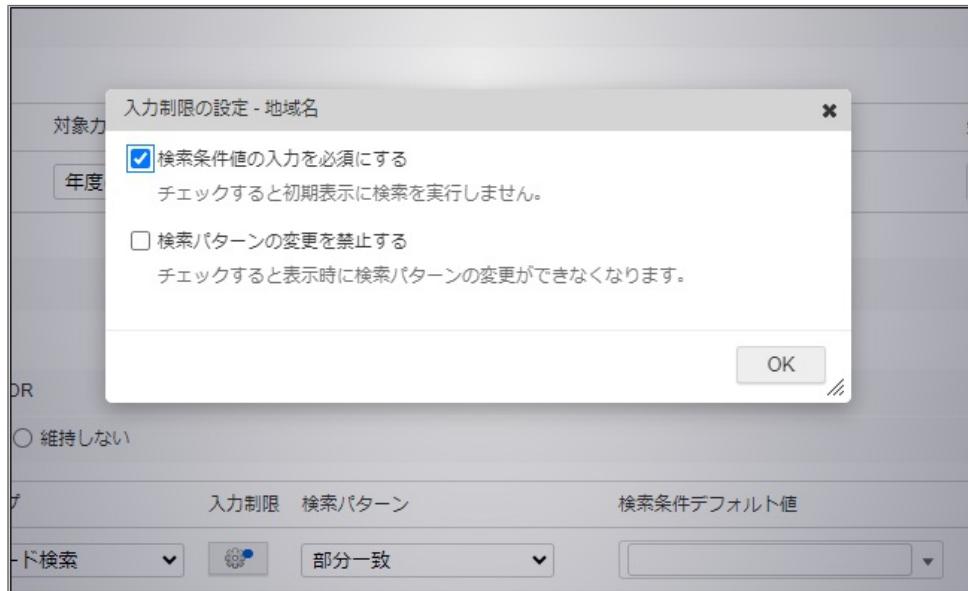
クリア 検索

i 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。

i 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。

i コラム

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

i コラム

クロス集計では、常に「検索」がツールバーに表示されます。



コラム

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

The screenshot shows the 'Search Type' and 'Default Value' sections of the search configuration. In the 'Default Value' field, the value '山' is entered. Below the input field, a dropdown menu lists various dynamic parameters:

- <%ENCODING%>
- <%LOCALE%>
- <%REQUEST_PARAMETER%>
- <%REQUEST_PARAMETER(KEY_NAME)%>
- <%SYSTEM_DATE%>
- <%SYSTEM_DATE(yyyy/MM/dd HH:mm)%>
- <%SYSTEM_DATE(yyyy/MM/dd)%>
- <%SYSTEM_DATE_TIMEZONE(yyyy/MM/dd HH:mm)%>
- <%USER_ID%>

At the bottom of the interface, there is a toolbar with various icons for text styling.



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

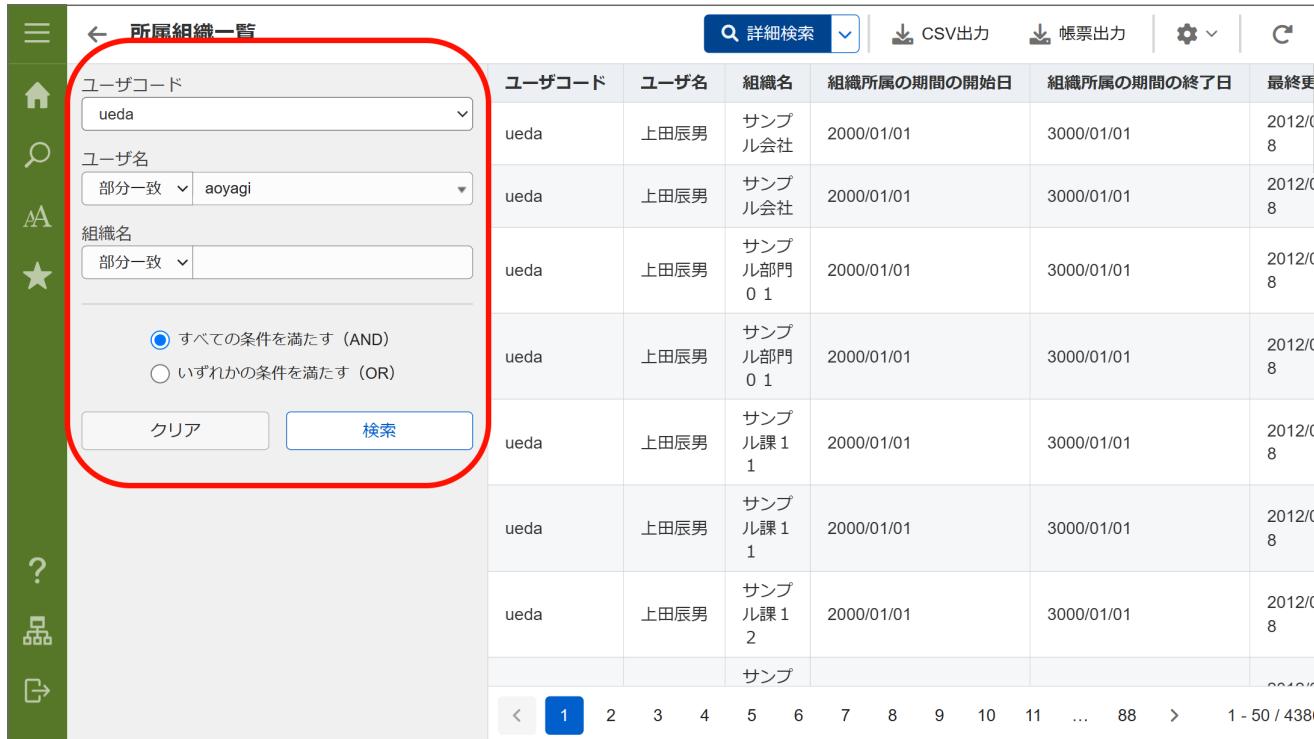
検索の条件値入力に選択型コンポーネントを利用する

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。



所属組織一覧

検索条件

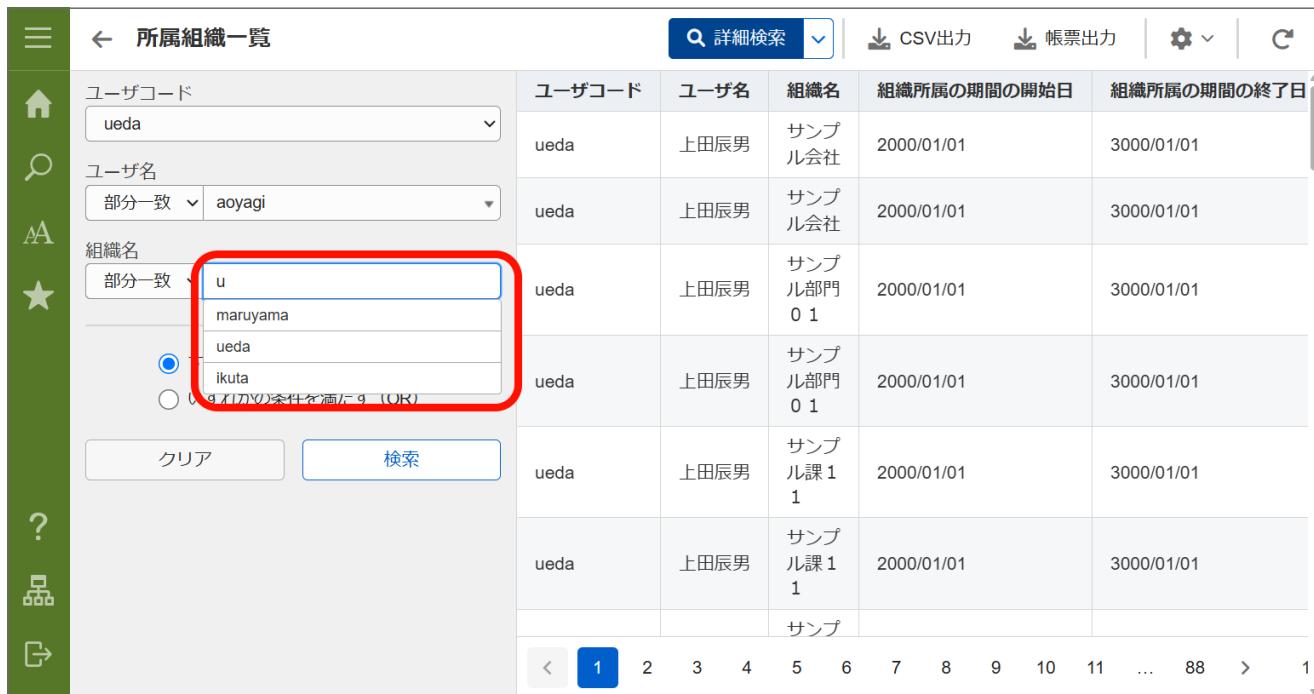
- ユーザコード: ueda
- ユーザ名: 部分一致 aoyagi
- 組織名: 部分一致
- 検索オプション: すべての条件を満たす (AND) いずれかの条件を満たす (OR)

検索ボタン: クリア 検索

ユーザコード	ユーザ名	組織名	組織所属の期間の開始日	組織所属の期間の終了日	最終更新日
ueda	上田辰男	サンブル会社	2000/01/01	3000/01/01	2012/08
ueda	上田辰男	サンブル会社	2000/01/01	3000/01/01	2012/08
ueda	上田辰男	サンブル部門01	2000/01/01	3000/01/01	2012/08
ueda	上田辰男	サンブル部門01	2000/01/01	3000/01/01	2012/08
ueda	上田辰男	サンブル課11	2000/01/01	3000/01/01	2012/08
ueda	上田辰男	サンブル課11	2000/01/01	3000/01/01	2012/08
ueda	上田辰男	サンブル課12	2000/01/01	3000/01/01	2012/08
		サンブル			2012/08

ページナビゲーション: < 1 2 3 4 5 6 7 8 9 10 11 ... 88 > 1 - 50 / 4386

オートコンプリートを利用した場合の例。



所属組織一覧

検索条件

- ユーザコード: ueda
- ユーザ名: 部分一致 aoyagi
- 組織名: 部分一致 u
- 検索オプション: すべての条件を満たす (AND) いずれかの条件を満たす (OR)

検索ボタン: クリア 検索

ユーザコード	ユーザ名	組織名	組織所属の期間の開始日	組織所属の期間の終了日
ueda	上田辰男	サンブル会社	2000/01/01	3000/01/01
ueda	上田辰男	サンブル会社	2000/01/01	3000/01/01
ueda	上田辰男	サンブル部門01	2000/01/01	3000/01/01
ueda	上田辰男	サンブル部門01	2000/01/01	3000/01/01
ueda	上田辰男	サンブル課11	2000/01/01	3000/01/01
ueda	上田辰男	サンブル課11	2000/01/01	3000/01/01
		サンブル		

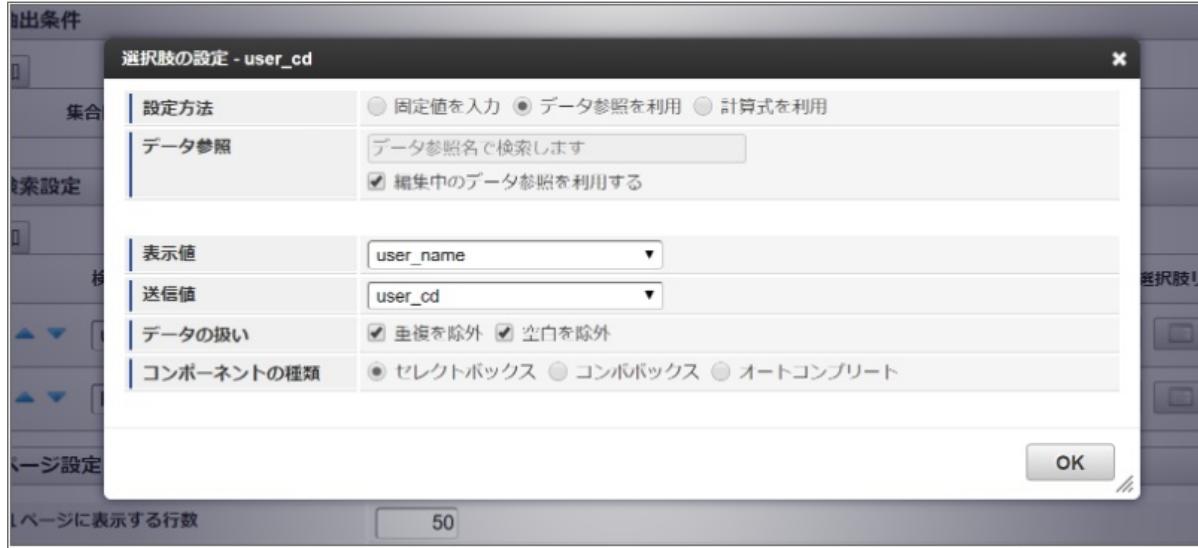
ページナビゲーション: < 1 2 3 4 5 6 7 8 9 10 11 ... 88 > 1

■ 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事も可能です。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

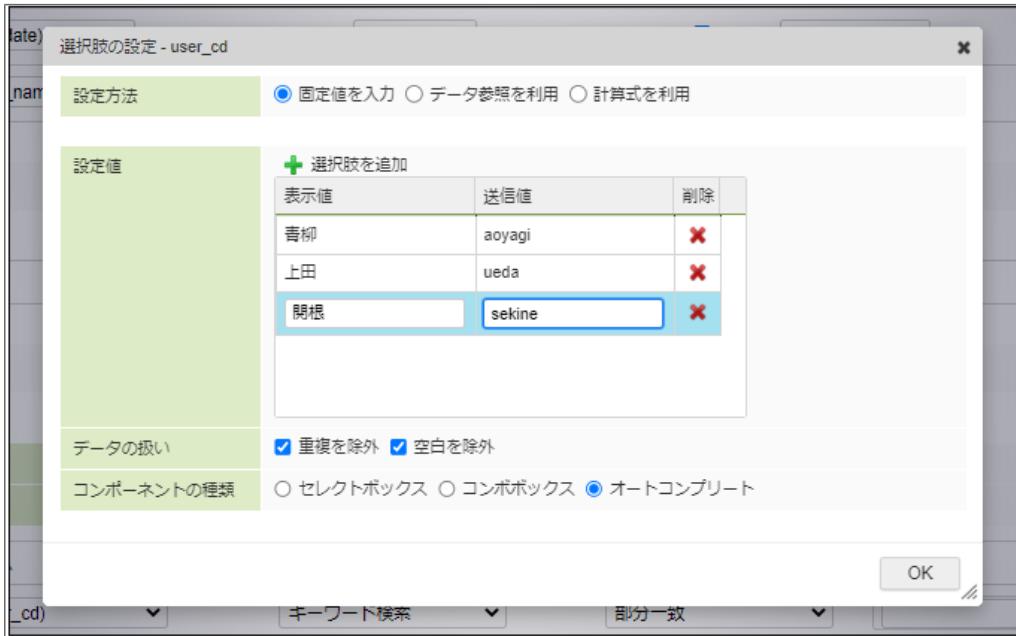
選択肢の表示値（ラベル）として利用するデータを取得するカラムを選択します。

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

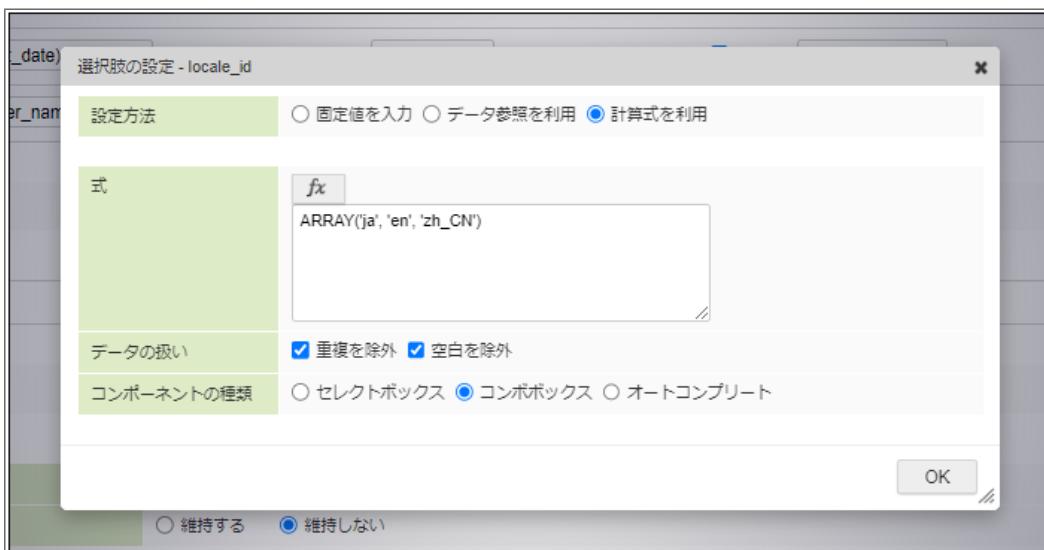
「表示値」と「送信値」を手入力で設定します。



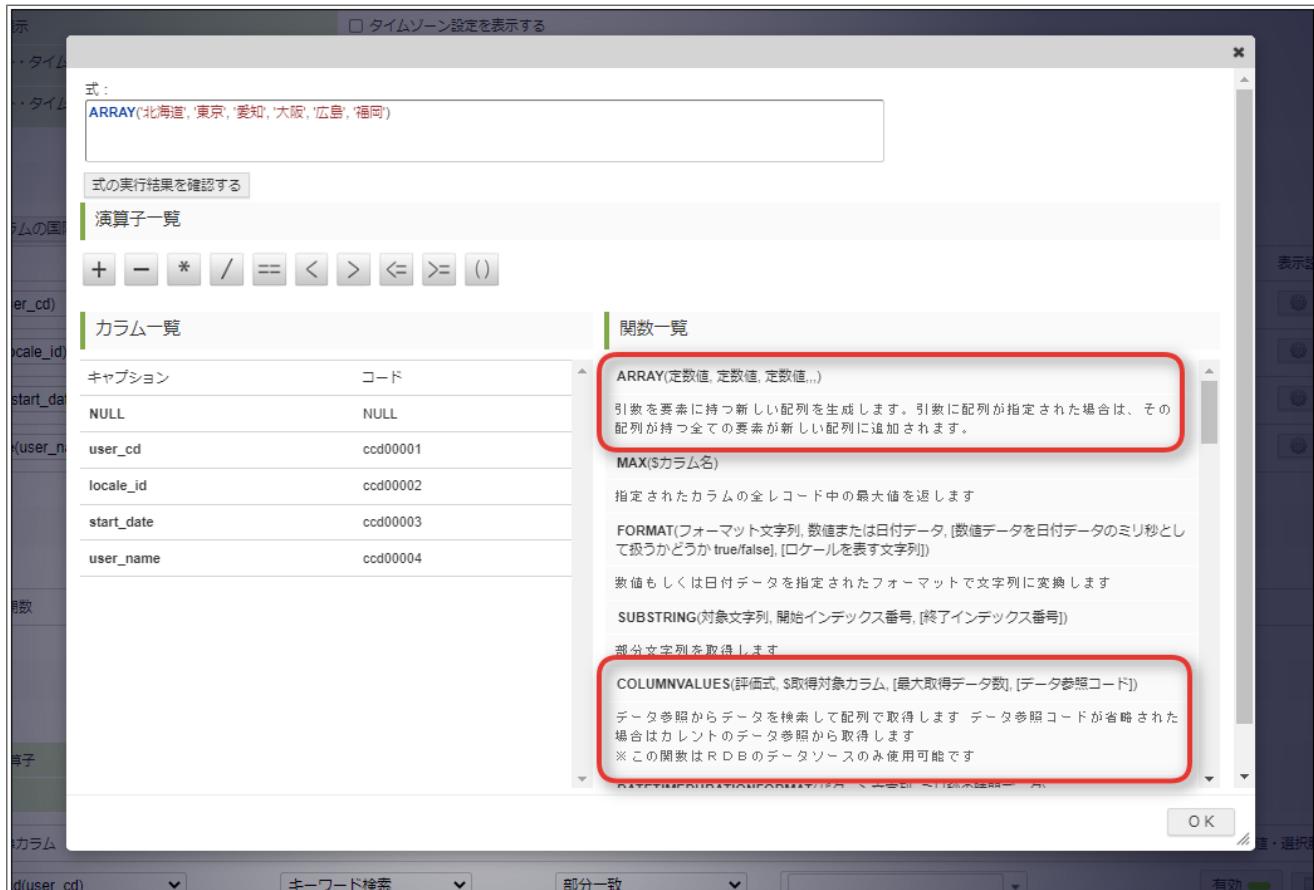
- 「計算式を利用」が選択された場合の設定

「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できます。

年度・都道府県別人口 - クロス集計

検索

入れ替え可能なカラム一覧 面積	列見出し 年度 ソート指定無し 年齢 ソート指定無し
行見出し 地域名 見出し名 ソート指定無し 都道府県名 見出し名 ソート指定無し	値 人口 合計 数値フォーマット

表示非表示 合計 平均 最小 最大 カウント

各項目を結合する演算子 AND OR

地域名	<input checked="" type="checkbox"/> すべて選択 <input checked="" type="checkbox"/> 北海道 <input checked="" type="checkbox"/> 東北 <input checked="" type="checkbox"/> 関東 <input checked="" type="checkbox"/> 中部 <input checked="" type="checkbox"/> 近畿 <input checked="" type="checkbox"/> 中国 <input checked="" type="checkbox"/> 四国 <input checked="" type="checkbox"/> 九州
人口	範囲指定
年度	範囲指定 1990 -

表示 クリア

		1990			1995			2000			合計	平均
北海道	北海道	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	小計	1,034	3,925	675	899	3,943	845	792	3,833	1,032	16,978	1,886.44
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		129	1,535	267	304	1,596	338	354	1,602	409	6,934	770.44

■ 設定方法

データ参照編集画面で、「検索タイプ」を「リスト選択」に設定してください。

検索設定

追加

各項目を結合する演算子	<input checked="" type="radio"/> AND <input type="radio"/> OR			
参照画面の操作状態	<input type="radio"/> 維持する <input checked="" type="radio"/> 維持しない			
検索対象カラム	検索タイプ	入力制限	検索パターン	検索条件デフォルト値
地域名(name)	リスト選択			
人口(population)	範囲検索	範囲指定		
年度(year)	範囲検索	範囲指定		

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。



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。

ページ設定

ページ設定

1ページに表示する行数	<input type="text" value="50"/>
ヘッダ	fx [ツールバー]
フッタ	fx [ツールバー]
▶ 國際化データ	

カテゴリ	項目名	説明
ページ設定	1ページに表示する行数	設定された行数（レコード数）ごとにページングします。
	ヘッダ	ヘッダ部に表示する文字列を設定できます。
	フッタ	フッタ部に表示する文字列を設定できます。

権限設定

権限設定ではデータ参照の参照権限を設定できます。

条件によって編集画面の権限設定は以下の様に表示されます。

- アクセス種別が選択できる場合

権限設定

アクセス種別	<input checked="" type="radio"/> 参照権 <input type="radio"/> 認可
参照権	<input type="button" value="追加"/>
<input type="checkbox"/> ゲストユーザ <input type="checkbox"/> 認証済みユーザ	

権限設定

アクセス種別	<input type="radio"/> 参照権 <input checked="" type="radio"/> 認可
認可	<small>① 認可の権限設定は登録／更新した後に設定できます。</small>

- アクセス種別が選択できない場合

権限設定

参照権	<input type="button" value="追加"/>
<input type="checkbox"/> ゲストユーザ <input type="checkbox"/> 認証済みユーザ	

標準では「テナント管理者」または「ViewCreator管理者」ロールを割り当てられたユーザがアクセス種別を選択できます。

任意のユーザにアクセス種別を選択させたい場合は「[アクセス種別-認可を利用する場合](#)」を参考に権限を設定してください。

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。

権限設定

参照権

	追加
<input type="checkbox"/>	盲
<input type="checkbox"/>	青柳辰巳
<input type="checkbox"/>	認可管理者
<input type="checkbox"/>	サンプル課 1 1
<input type="checkbox"/>	サンプル課 2 1
<input type="checkbox"/>	パブリックグループa
<input type="checkbox"/>	パブリックグループc

ゲストユーザ 認証済みユーザ

[登録して一覧へ戻る](#)

i コラム

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

参照権は削除したい設定にチェックを入れてゴミ箱アイコンをクリックすることで削除できます。

権限設定

参照権

	追加
<input type="checkbox"/>	盲
<input checked="" type="checkbox"/>	青柳辰巳
<input type="checkbox"/>	認可管理者
<input type="checkbox"/>	サンプル課 1 1
<input type="checkbox"/>	サンプル課 2 1
<input type="checkbox"/>	パブリックグループa
<input type="checkbox"/>	パブリックグループc

ゲストユーザ 認証済みユーザ

[登録して一覧へ戻る](#)

設定が終わったら登録ボタンをクリックします。

権限設定

参照権

	追加
<input type="checkbox"/>	盲
<input type="checkbox"/>	青柳辰巳
<input type="checkbox"/>	認可管理者
<input type="checkbox"/>	サンプル課 1 1
<input type="checkbox"/>	サンプル課 2 1
<input type="checkbox"/>	パブリックグループa
<input type="checkbox"/>	パブリックグループc

ゲストユーザ 認証済みユーザ

[登録して一覧へ戻る](#)

! 注意

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

i コラム

データ参照を閲覧するためには、認可の設定も必要です。

必要に応じて設定を追加、変更してください。

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

認可の設定

認可を選択してデータ参照を登録／更新すると、データ参照名の認可リソースが自動的に作成されます。

認可リソース作成後は認可設定画面で役職や下位組織などの条件で参照権限を設定できます。

認可利用時の注意事項については「[アクセス種別-認可の利用について](#)」を参照してください。

i コラム

認可の設定は2025 Spring(Kamille)から利用可能です。

権限設定

アクセス種別	<input type="radio"/> 参照権 <input checked="" type="radio"/> 認可
認可	① 認可の権限設定は登録／更新した後に設定できます。

認可設定画面は登録更新後のデータ参照一覧のツールバー、またはデータ参照・編集画面から表示できます。

認可の設定方法については、「[テナント管理者操作ガイド](#)」 - 「[認可を設定する](#)」を参照してください。

■ 一覧

The screenshot shows a 'Data Reference List' screen. At the top, there is a toolbar with various icons: Copy, Delete, Source Code Output, CSV Job Registration, Portlet Registration, and Access Rights (which is highlighted with a red box). Below the toolbar is a search bar with '検索' (Search) and 'クリア' (Clear) buttons. The main area is a table listing data references:

	編集	データ参照作成	データ参照名	更新日	作成者	エクスポート
<input type="checkbox"/>			2000年度版・地域 - 都道府県 サマリ集計	2025/01/29 4:17	tenant	
<input type="checkbox"/>			2000年度版・地域 - 都道府県 サマリ集計/面積順	2013/10/31 4:20	tenant	
<input type="checkbox"/>			2010年度版・地域 - 都道府県 サマリ集計	2020/02/06 15:47	tenant	
<input type="checkbox"/>			2010年度版・地域 - 都道府県 サマリ集計/面積順	2020/02/06 15:38	tenant	
<input type="checkbox"/>			サマリ集計から作成した標準データ	2013/11/06 2:30	tenant	

■ 参照・編集画面

The screenshot shows the 'Access Rights' configuration screen. It has a toolbar at the top with 'Copy', 'Delete', 'Source Code Output', 'CSV Job Registration', 'Portlet Registration', and 'Access Rights'. Below the toolbar is a table with two rows:

アクセス種別	<input type="radio"/> 参照権 <input checked="" type="radio"/> 認可
認可	vc-dataview://5i7urzmesvqqip-

i コラム

アクセス種別-認可のデータ参照画面の表示にはURL「viewcreator/dataview/{データ参照の種類}_view/{データ参照コード}」を使用できます。

このURLをメニューに設定すると、参照権限がないユーザはメニューに表示されなくなります。

データ参照の種類については「[データ参照へのショートカット](#)」を参照してください。

i コラム

アクセス種別-認可のデータ参照は管理者、作成者でも認可設定画面で参照権限を付与する必要があります。

「[アクセス種別-認可の利用について](#)」を参照してください。

ここでは詳細機能についてご紹介します。

抽出条件の設定

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

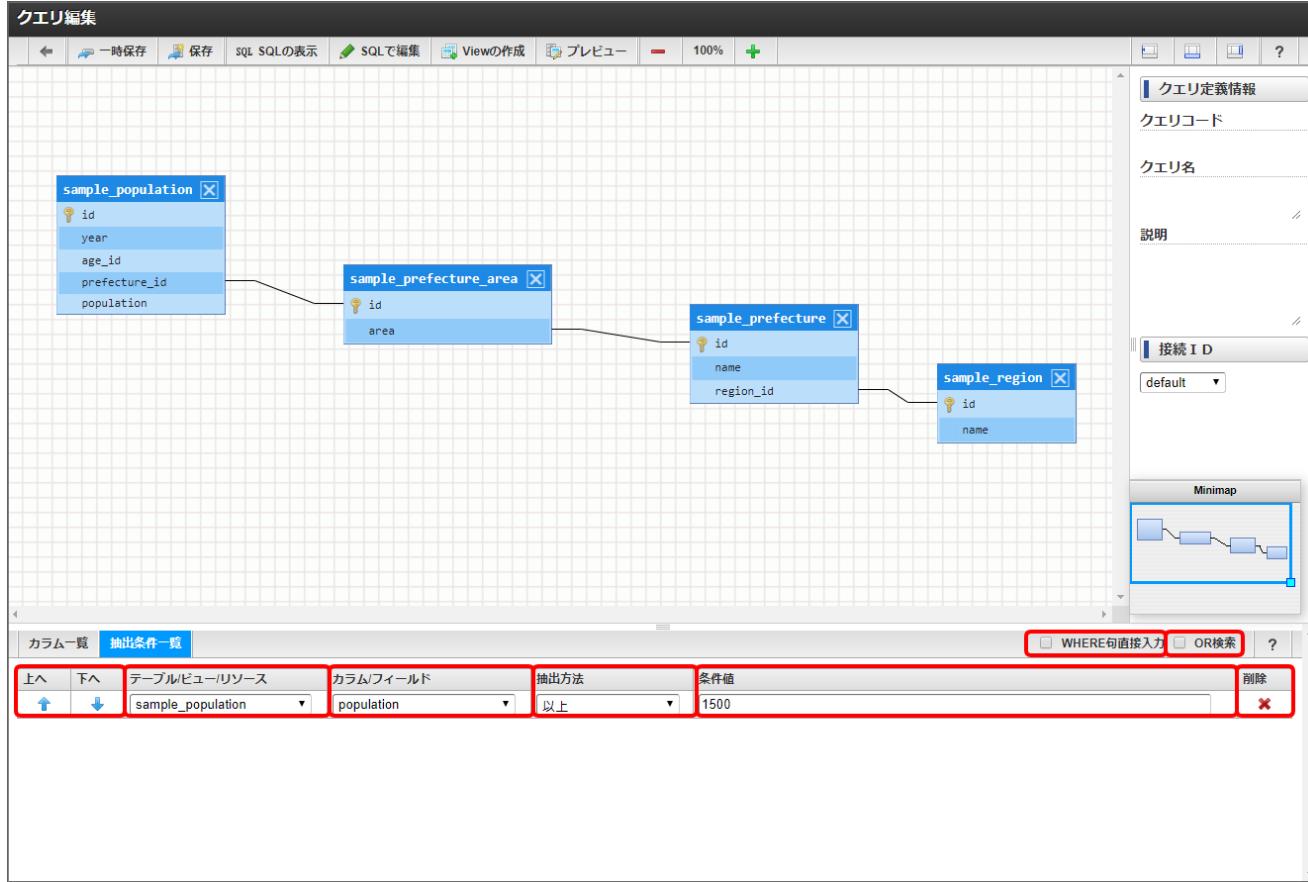
リスト集計の「検索設定」は、データ参照を閲覧するユーザが目的の情報に素早く辿り着くための機能ですが、それに対して「抽出条件」は、ユーザに見せたくない情報を設定者側であらかじめ絞っておくための機能です。

目次

- クエリ作成時に抽出条件を設定する
 - 抽出条件を直接入力する
- データ参照作成時に抽出条件を設定する
- データ参照時のSQL
- 動的パラメータ
 - <%DYNAMIC_PARAMETER%>
 - <%ENCODING%>
 - <%LOCALE%>
 - <%NOT_NULL%>
 - <%NULL%>
 - <%SYSTEM_DATE(*)%>
 - <%USER_ID%>
 - <%REQUEST_PARAMETER(KEY_NAME)%>
 - <%REQUEST_PARAMETER_VALUES(KEY_NAME)%>
 - <%REQUEST_PARAMETER_NUMBER_VALUES(KEY_NAME)%>
 - <%TIME_ZONE_ID%>
 - オリジナルの動的パラメータ作成
 - 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。
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。

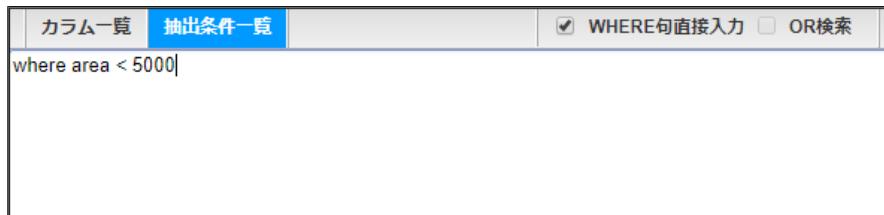
抽出条件を直接入力する

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。

項目名	説明
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

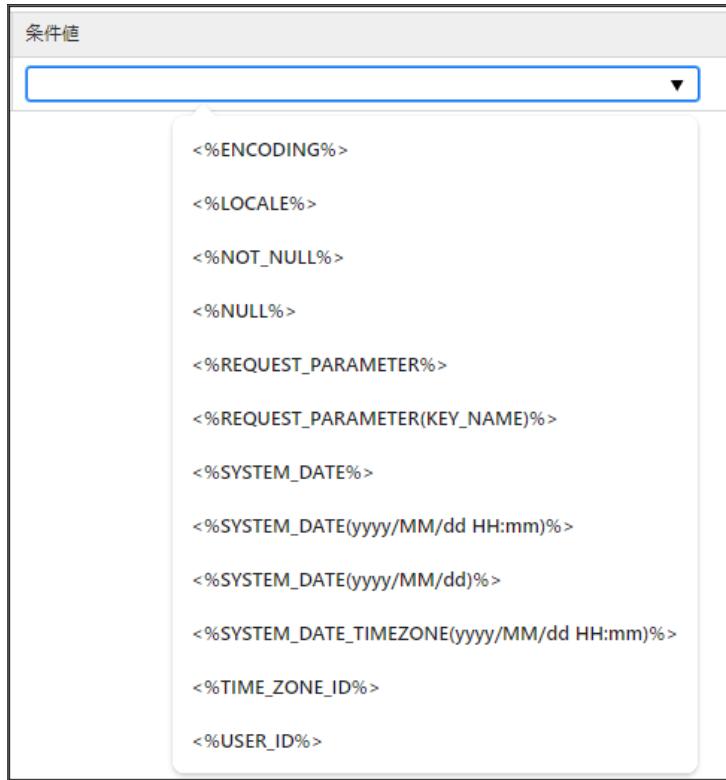
データ参照が閲覧されたときに発行されるSQLは次のような構造です。

```
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

<%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。

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。



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。

<%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)%>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。

ここでは例として「都道府県名」の検索値として渡す値のリクエストパラメータのキー名を param としています。

The screenshot shows the '検索設定' (Search Settings) panel. At the top, there's a placeholder text box containing '<%REQUEST_PARAMETER(param)%>'. Below it are sections for '検索対象カラム' (Search Target Column), '検索タイプ' (Search Type), '検索条件デフォルト値' (Default Search Condition Value), and '検索条件値・選択肢リスト' (Search Condition Value · Selection List). Two search conditions are defined: one for '都道府県名(name)' using '部分一致' (Partial Match) and another for '面積(area)' using '範囲検索' (Range Search).

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。

例では、以下のようなURLにアクセスすることで「都道府県名」の検索値として「島」を指定した際と同じ結果を表示しています。

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

The screenshot shows the '2000年度版・地域・都道府県 サマリ集計' (2000 Annual Edition · Region · Prefecture Summary Collection) page. A search bar at the top contains the value '島' (Ishikawa Prefecture), which is highlighted with a red box. The main area displays a table of regional statistics. The table has columns: 地域名 (Region Name), 都道府県名 (Prefecture Name), 面積(平均) (Average Area), and 人口(合計) (Total Population). The data shows Ishikawa Prefecture (島根) with an average area of 6,708 and a total population of 761.

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

2. 抽出条件の場合

タグが受け取る引数として、リクエストパラメータのキー名を任意に指定します。

ここでは例として「地域名」の抽出条件値として渡す値のリクエストパラメータのキー名を `region`、「都道府県名」のキー名を `pref` としています。

The screenshot shows the '抽出条件' (Extraction Conditions) panel. It contains two entries under the '条件値' (Condition Value) column, each with a placeholder text box containing '<%REQUEST_PARAMETER()%>'. The first entry is for '地域名(name)' and the second is for '都道府県名(name)'.

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。

例では、以下のようなURLにアクセスすることで「地域名」の抽出条件値として「関東」、「都道府県名」の抽出条件値として「東京」を指定した際と同じ結果を表示しています。

`.../viewcreator/dataview/list/{dataview_cd}?region=関東&pref=東京`

2000年度版・地域 - 都道府県 サマリ集計					
地域名	都道府県名	面積(平均)	人口(合計)		詳細
関東	東京	2,102	11,409	★	
関東	東京	2,102	11,656	★	
関東	東京	2,102	11,598	★	
関東	東京	2,102	11,819	★	
関東	東京	2,102	11,762	★	
関東	東京	2,102	11,735	★	
関東	東京	2,102	12,017	★	
		2,102	81,996		

3. ポートレットで利用する場合

メニューから「サイトマップ」→「ポータル - ポートレット一覧」を選択します。

編集アイコンをクリックし、「ポートレット編集」画面を開きます。

基本設定のページ引数に抽出条件で設定したタグのキー名を指定します。

基本設定							
ポートレット名	スクリプト開発モデルポートレット						
名称*	<table border="1"> <tr> <td>日本語*</td> <td>2000年度版・地域 - 都道府県 サマリ集計</td> </tr> <tr> <td>英語*</td> <td>2000-Regions-Prefectures summary</td> </tr> <tr> <td>中国語(中華人民共和国)*</td> <td>2000年度版本 区域 - 都道府县 统计</td> </tr> </table>	日本語*	2000年度版・地域 - 都道府県 サマリ集計	英語*	2000-Regions-Prefectures summary	中国語(中華人民共和国)*	2000年度版本 区域 - 都道府县 统计
日本語*	2000年度版・地域 - 都道府県 サマリ集計						
英語*	2000-Regions-Prefectures summary						
中国語(中華人民共和国)*	2000年度版本 区域 - 都道府县 统计						
アプリケーション	<table border="1"> <tr> <td>日本語</td> <td>ViewCreator</td> </tr> <tr> <td>英語</td> <td>ViewCreator</td> </tr> <tr> <td>中国語(中華人民共和国)</td> <td>ViewCreator</td> </tr> </table>	日本語	ViewCreator	英語	ViewCreator	中国語(中華人民共和国)	ViewCreator
日本語	ViewCreator						
英語	ViewCreator						
中国語(中華人民共和国)	ViewCreator						
ページパス*	viewcreator/views/dataview/list4portal						
ページ汎用	portlet=true&dataview_cd=5i7urzmesvqqip4®ion=関東						
Actionハンドラ	<input type="checkbox"/>						
Eventハンドラ	<input type="checkbox"/>						

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

2000年度版・地域 - 都道府県 サマリ集計						
年度	地域名	都道府県名	年齢	面積(平均)	人口(合計)	
1975	関東	東京	○	2,102	11,656	★
1995	関東	東京	○	2,102	11,735	★
1980	関東	東京	○	2,102	11,598	★
2000	関東	東京	○	2,102	12,017	★
1985	関東	東京	○	2,102	11,819	★
1970	関東	東京	○	2,102	11,409	★
1990	関東	東京	○	2,102	11,762	★
1970	関東	神奈川	○	2,416	5,473	★
1995	関東	神奈川	○	2,416	8,238	★
1980	関東	神奈川	○	2,416	6,019	★
1975	関東	神奈川	○	2,416	6,394	★
1990	関東	神奈川	○	2,416	7,955	★
1985	関東	神奈川	○	2,416	7,429	★
2000	関東	神奈川	○	2,416	8,475	★
1980	関東	埼玉	○	3,767	5,417	★
1975	関東	埼玉	○	3,767	4,819	★
1995	関東	埼玉	○	3,767	6,748	★
1970	関東	埼玉	○	3,767	3,867	★

<%REQUEST_PARAMETER_VALUES(KEY_NAME)%>

HttpServletRequest（リクエスト）、または、RenderRequest（ポートレットのリクエスト）から取得した同名の複数パラメータを配列として取得し、カンマ区切りの値としてセットされます。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。

複数選択のチェックボックスや複数選択リストなどで送信される値を一括でIN句に利用する場合に使用します。

i コラム

- クエリビルダのWHERE句直接入力と抽出方法「直接入力」、SQLビルダで利用可能です
- 該当するリクエストパラメータが存在しない場合は「NULL」に変換されます
- 最大1000個までの要素を処理します

i コラム

<%REQUEST_PARAMETER_VALUES(KEY_NAME)%> は2025 Autumn(Lilac)から利用可能です。

i コラム

使用例

基本的な使用例

複数の文字列値をIN句で指定する場合：

```
SELECT * FROM imm_user WHERE user_cd IN (<%REQUEST_PARAMETER_VALUES(userCdList)%>)
```

リクエストパラメータとSQL変換

リクエストパラメータ「userCdList=aoyagi&userCdList=ueda&userCdList=harada」が送信されると、以下のSQLが実行されます：

```
SELECT * FROM imm_user WHERE user_cd IN ('aoyagi', 'ueda', 'harada')
```

空文字の処理

空文字は無視されます。

リクエストパラメータ「userCdList=aoyagi&userCdList=&userCdList=ueda」が送信された場合：

```
SELECT * FROM imm_user WHERE user_cd IN ('aoyagi', 'ueda')
```

<%REQUEST_PARAMETER_NUMBER_VALUES(KEY_NAME)%>

HttpServletRequest（リクエスト）、または、RenderRequest（ポートレットのリクエスト）から取得した同名の複数パラメータを数値配列として取得し、カンマ区切りの数値としてセットされます。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。

数値の複数選択や範囲指定などで送信される値を一括でIN句に利用する場合に使用します。

i コラム

- クエリビルダのWHERE句直接入力と抽出方法「直接入力」、SQLビルダで利用可能です
- 該当するリクエストパラメータが存在しない場合や、数値変換できる値がない場合は「NULL」に変換されます
- 数値変換できない値は無視され、警告ログが出力されます
- 最大1000個までの要素を処理します

i コラム

<%REQUEST_PARAMETER_NUMBER_VALUES(KEY_NAME)%> は2025 Autumn(Lilac)から利用可能です。



コラム

使用例

基本的な使用例

複数の数値をIN句で指定する場合：

```
SELECT * FROM imm_user WHERE sort_key IN (<%REQUEST_PARAMETER_NUMBER_VALUES(sortKeys)%>)
```

リクエストパラメータとSQL変換

リクエストパラメータ「sortKeys=1&sortKeys=2&sortKeys=3」が送信されると、以下のSQLが実行されます：

```
SELECT * FROM imm_user WHERE sort_key IN (1, 2, 3)
```

データ変換の処理

数値変換できない値や空文字は無視されます。

リクエストパラメータ「sortKeys=1&sortKeys=&sortKeys=abc&sortKeys=2」が送信された場合：

```
SELECT * FROM imm_user WHERE sort_key IN (1, 2)
```

<%TIME_ZONE_ID%>

- ログインユーザのタイムゾーンIDがセットされます。下記の様にパラメータを追加することで、他の種類のタイムゾーンIDのセットもできます。
 - <%TIME_ZONE_ID(SYSTEM)%> : システムのタイムゾーンIDがセットされます。
 - <%TIME_ZONE_ID(TENANT)%> : テナントのタイムゾーンIDがセットされます。
 - <%TIME_ZONE_ID(USER)%> : ログインユーザのタイムゾーンIDがセットされます。パラメータ省略時と同じです。



コラム

<%TIME_ZONE_ID%>は2024 Autumn(Jasmine)から利用可能です。

オリジナルの動的パラメータ作成

下記の設定ファイルでオリジナルの動的パラメータを追加できます。

設定ファイル: WEB-INF/conf/viewcreator-config.xml

パラメータ名の制約: <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。

単一値パラメータの設定と実装

单一の値を返すパラメータを作成する場合は、jp.co.intra_mart.foundation.viewcreator.SystemParameterインターフェースを実装します。

設定例（ログインユーザのロケール情報を取得するパラメータ）：

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

実装例（ログインユーザのロケール文字列を返す実装）：

```
public class SystemParameterLocale implements jp.co.intra_mart.foundation.viewcreator.SystemParameter {

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

複数の値を配列として返すパラメータを作成する場合は、jp.co.intra_mart.foundation.viewcreator.ArraySystemParameterインターフェースを実装します。

設定例（ログインユーザが所属するロールIDの配列を取得するパラメータ）：

```
<param>
<param-name>SAMPLE_PARAM_ROLE_IDS</param-name>
<class-name>sample.SampleParameterRoleIds</class-name>
</param>
```

実装例（ログインユーザのロールIDの配列を返す実装）：

```
public class SampleParameterRoleIds implements jp.co.intra_mart.foundation.viewcreator.ArraySystemParameter {

    @Override
    public Object[] getArray(final String syskey) {
        final AccountContext context = Contexts.get(AccountContext.class);
        return context.getRoleIds().toArray(new String[0]);
    }
}
```

使用例：

```
-- ログインユーザのロールIDを使った抽出条件
SELECT * FROM b_m_role_b WHERE role_id IN (<%SAMPLE_PARAM_ROLE_IDS%>);

-- 実行時変換後（例）
SELECT * FROM b_m_role_b WHERE role_id IN ('tenant_manager', 'im_workflow_user');
```

追加したパラメータ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箇所

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	配列型パラメータ
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 利用可能
		抽出条件（抽出方法が直接入力以外のとき）	自動変換 利用不可
		WHERE句直接入力	そのまま利用 利用可能
		Viewの作成	利用不可 利用不可
	SQLビルダ	SQL	そのまま利用 利用可能
データ参照	抽出条件	自動変換	利用不可
	検索設定	自動変換	利用不可

「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、シングルクオ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。

「配列型パラメータ」の列は、配列型動的パラメータが利用可能かを示します。

製品標準では <%REQUEST_PARAMETER_VALUES(KEY_NAME)%> と <%REQUEST_PARAMETER_NUMBER_VALUES(KEY_NAME)%> を提供します。

jp.co.intra_mart.foundation.viewcreator.ArraySystemParameter インタフェースを実装することで独自の配列型パラメータを追加することも可能です。



コラム

シングルクォーテーションの自動エスケープ

すべての動的パラメータについて、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）に自動変換されます。

例：

- 単一値の場合：受け取ったデータが「user's」の場合は「user''s」として処理
- 配列値の場合：受け取ったデータが「user's」「o'connor」の場合は「'user''s', 'o''connor'」として処理

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設定できます。

← 年度 - 地域 - 都道府県 サマリ集計（データ結合）				
年度	都道府県名	地域名	面積	人口
1970	福井	中部	4,189	67
1970	和歌山	近畿	4,726	243
1970	島根	中国	6,708	508
1970	茨城	関東	6,096	169
1970	石川	中部	4,185	82
1970	京都	近畿	4,613	484
1970	福島	東北	13,783	155
1970	大分	九州	5,805	280
1970	福岡	九州	4,843	943
1970	岐阜	中部	10,209	138
1970	山梨	中部	4,201	188
1970	青森	東北	9,235	940
1970	岐阜	中部	10,209	1,194

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

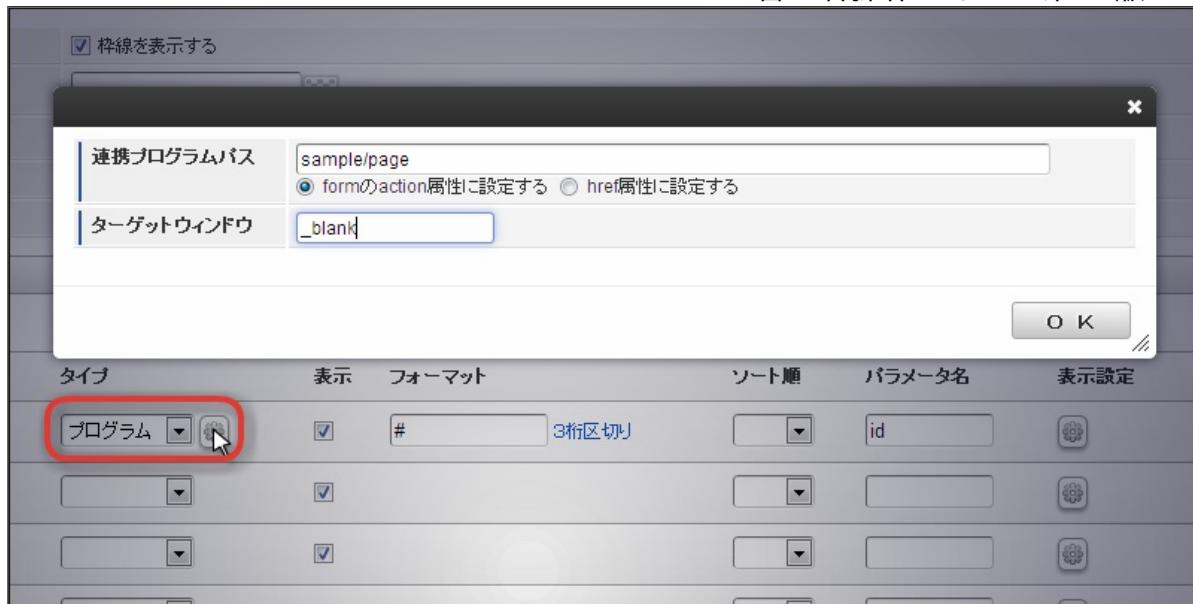
■ 設定方法

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。

← 年度 - 地域 - 都道府県 サマリ集計 (データ結合)					詳細検索	CSV出力	帳票出力	
年度	都道府県名	地域名	面積	人口				
1970	福井	中部	4,189	67				
1970	和歌山	近畿	4,726	243				
1970	島根	中国	6,708	508				
1970	茨城	関東	6,096	169				
1970	石川	中部	4,185	82				
1970	京都	近畿	4,613	484				
1970	福島	東北	13,783	155				
1970	大分	九州	5,805	280				
1970	福岡	九州	4,843	943				
1970	岐阜	中部	10,209	138				
1970	山梨	中部	4,201	188				
1970	青森	東北	9,235	940				
1970	岐阜	中部	10,209	1,194				

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。

画像の例では以下のようなデータが渡されます。

year	1970
prefecture_name	福井
region_name	中部
area	4189
population	67

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。

タイプ	表示	フォーマット	ソート順	パラメータ名	表示設定
▼ プログラム ▼	☑	#	3桁区切り	id	□
▼	☑				□
▼	☑				□
▼					□

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。



カラム

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。

連携プログラムパス	viewcreator/dataview/{データ参照} {ターンコード} {データ参照コード}
ターゲットウィンドウ	_blank
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	
コード(data_cd)	プログラム ▼
名(data_name)	プログラム ▼
名(国際化)(data_name)	プログラム ▼

- 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。

The screenshot shows the 'Data Reference List' screen. At the top, there is a toolbar with various icons: Copy (копировать), Delete (удалить), Source Code Output (Создать исходный код), CSV Output Job Registration (CSV-вывод задачи), Portlet Registration (Портал), and Go to Query List (Перейти в список запросов). The 'Portlet Registration' button is highlighted with a red box. Below the toolbar is a search bar with fields for 'Search' and 'Clear'. The main area is a table listing data references:

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

ダイアログが表示されるので、OKボタンをクリックします。

次に、「サイトマップ」→「ポータル - ポートレット一覧」を選択します。

ポートレット一覧

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

表示先ポータル種別 | アプリケーション | 検索 | ページ中 1 / 1 ページ目 | リスト | 10 | 16 件中 1 - 10 を表示

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

アクセス権設定									
アクションの種類		全てのアクション ▾							
リソース		アクション	認証		組織		ロール		
ゲストユーザ	認証済みユーザ	サンプル会社	その他会社	テナント管理者	認可管理者	メニュー管理者	メニュー一連用管理者	メニュー二連用管理者	
サマリ集計から作成した棒グラフ	利用 >								

適宜、適切なアクセス権を設定します。

The screenshot shows the intra-mart Accel Platform ViewCreator interface. At the top, there is a navigation bar with links for Top, Workflow, テナント管理 (Tenant Management), サンプル (Sample), and サイトマップ (Site Map). Below the navigation bar, there is a toolbar with icons for adding, back, forward, and search. A tab bar shows 'グループポータル' (Group Portal) and 'サンプル' (Sample). The main content area has a title 'リンク集' (Link Collection) and a section titled 'intra-martリンク' (intra-mart Links) containing three items: 'NTTデータ イントラマートのホームページ' (NTT DATA INTRAMART Home Page), 'intra-mart FAQ', and 'OPEN INTRA-MART'. Below this is another section titled 'サマリ集計から作成した棒グラフサンプル' (Bar Chart Sample from Summary Statistics) which displays a grouped bar chart comparing land area and population across various Japanese prefectures.

Prefecture	面積 (Area)	人口 (Population)
... (Oita)	~15,000	~5,500
埼玉 (Saitama)	~4,000	~4,000
千葉 (Chiba)	~5,000	~3,500
石川 (Ishikawa)	~4,500	~1,200
滋賀 (Shiga)	~4,000	~1,000
京都 (Kyoto)	~4,800	~2,500
兵庫 (Hyogo)	~8,500	~5,000
... (Fukuoka)	~5,000	~1,200
島根 (Ishigaki)	~7,000	~1,000

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

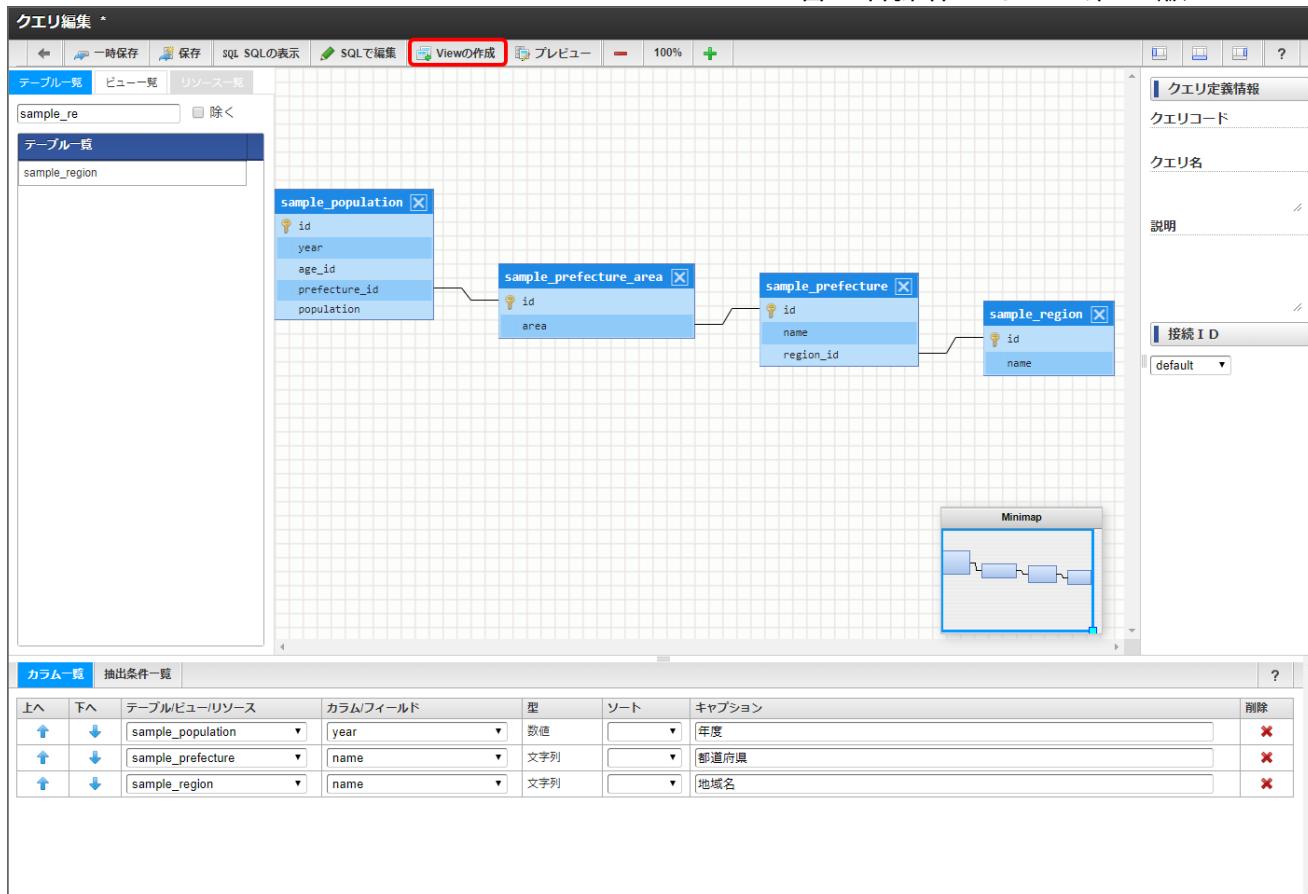
作成したクエリをベースにして、データベース上にViewを作成（Create View）できます。

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

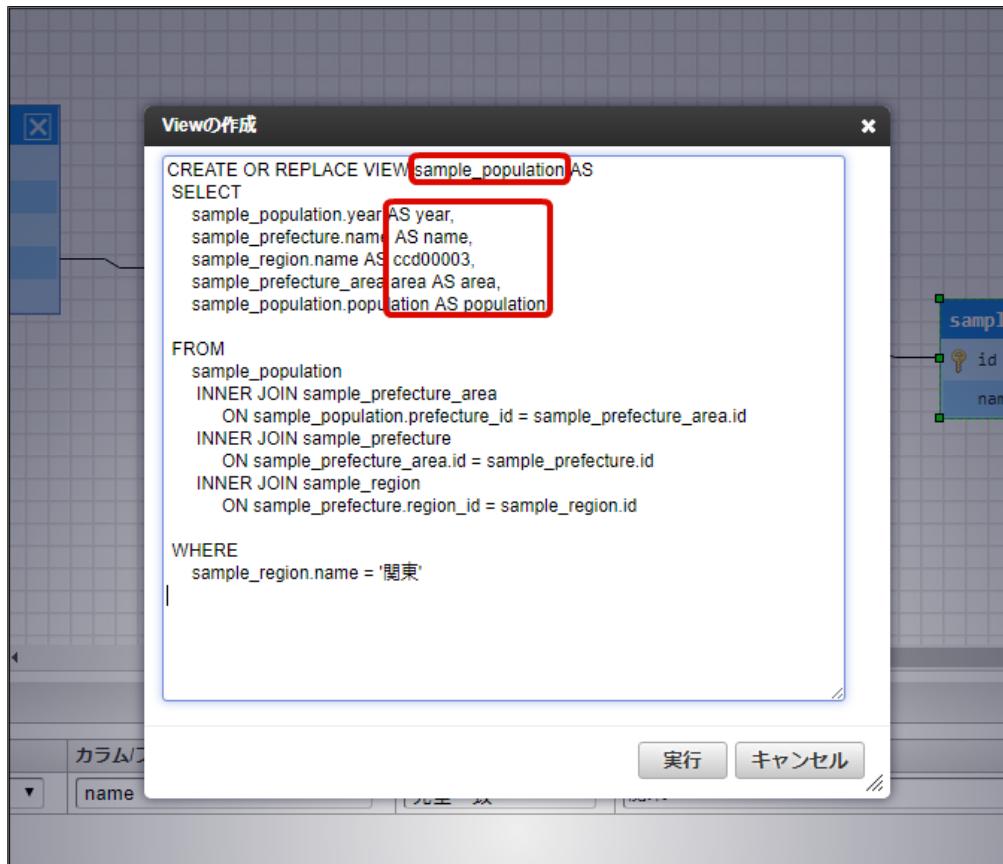
作成したViewはテーブルと同じように、クエリに組み込むことができます。

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できます。

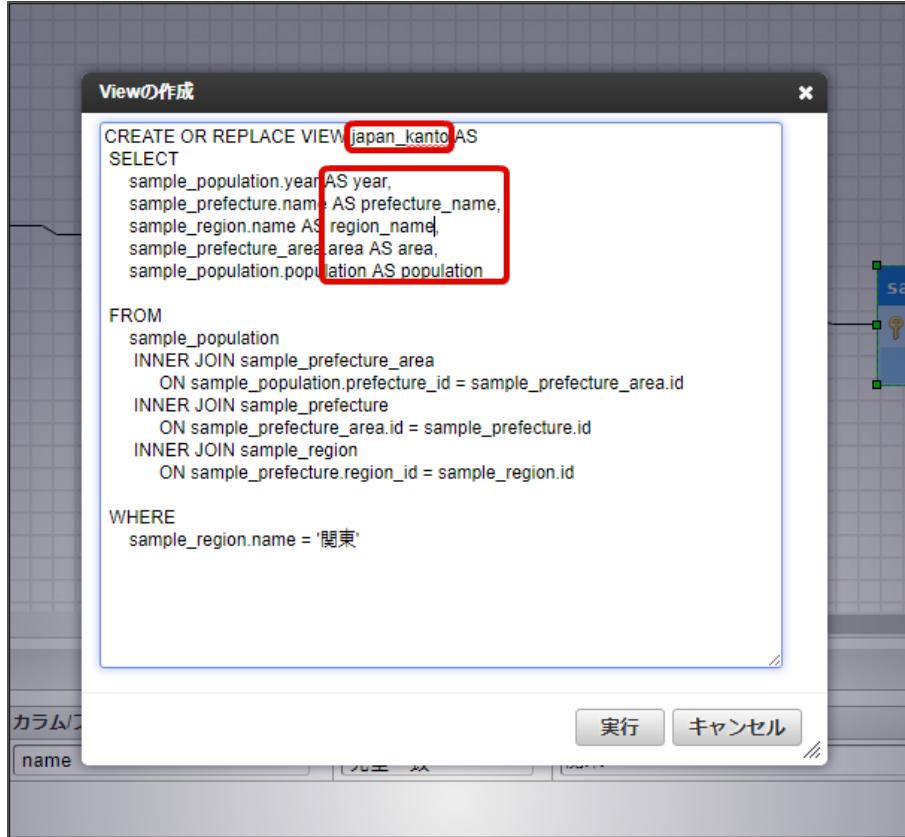
クエリ編集画面のツールバーから、「Viewの作成」をクリックします。



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。

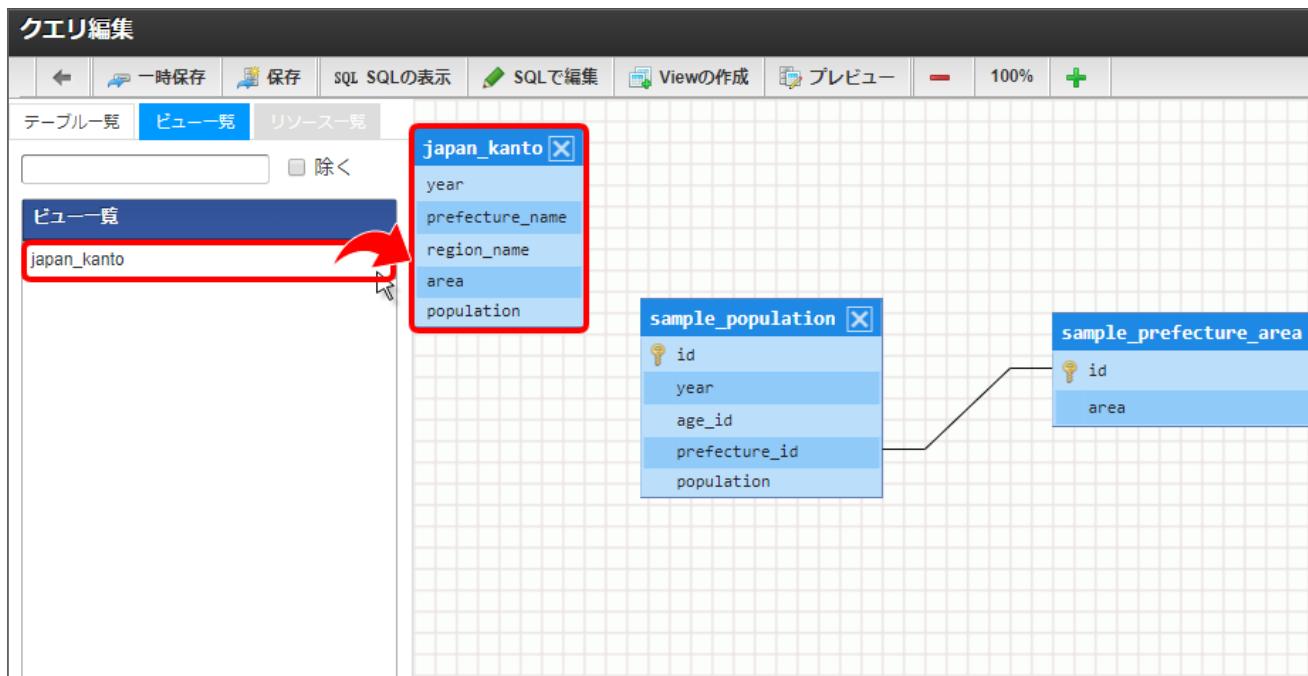


注意

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できます。

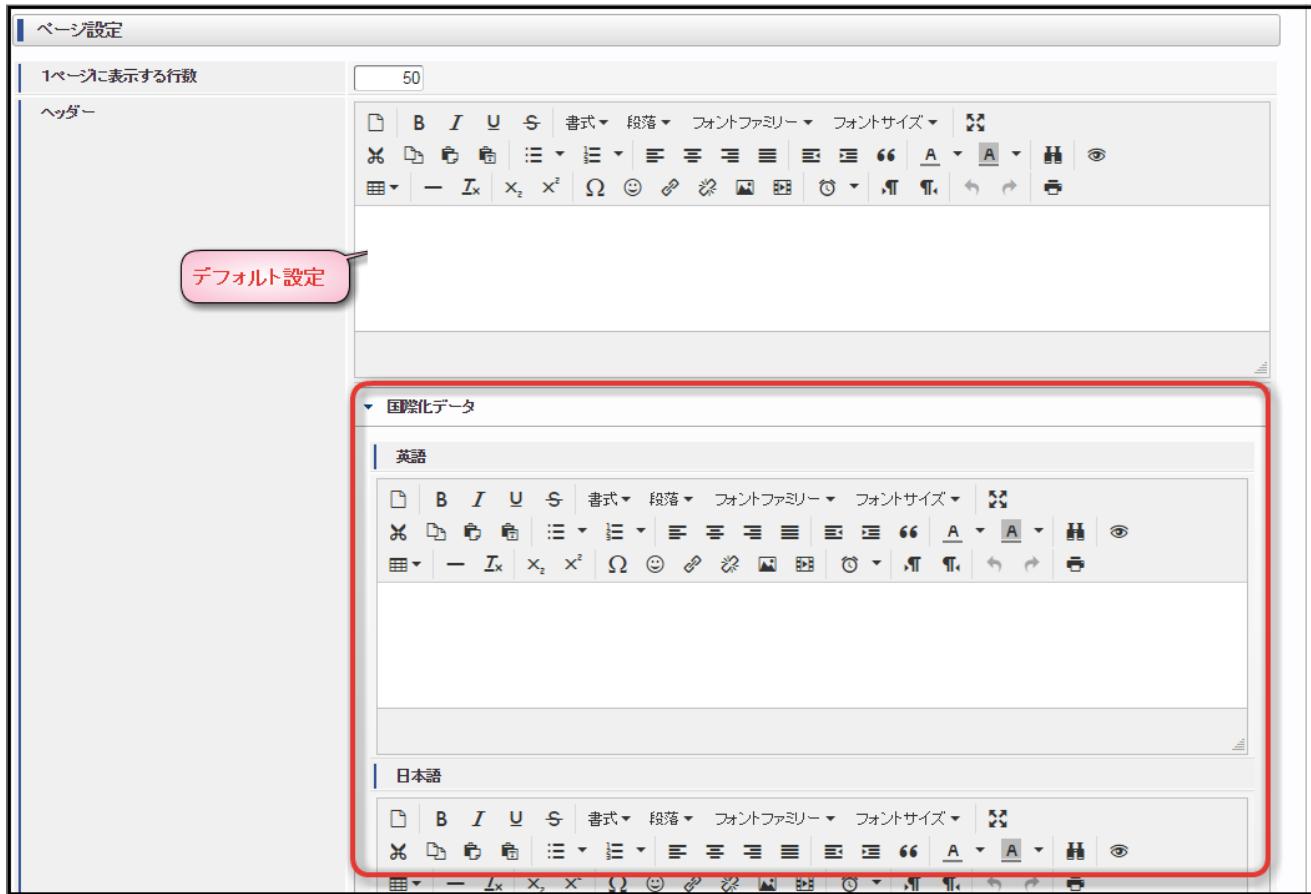
- データ参照名の国際化

「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。

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

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。



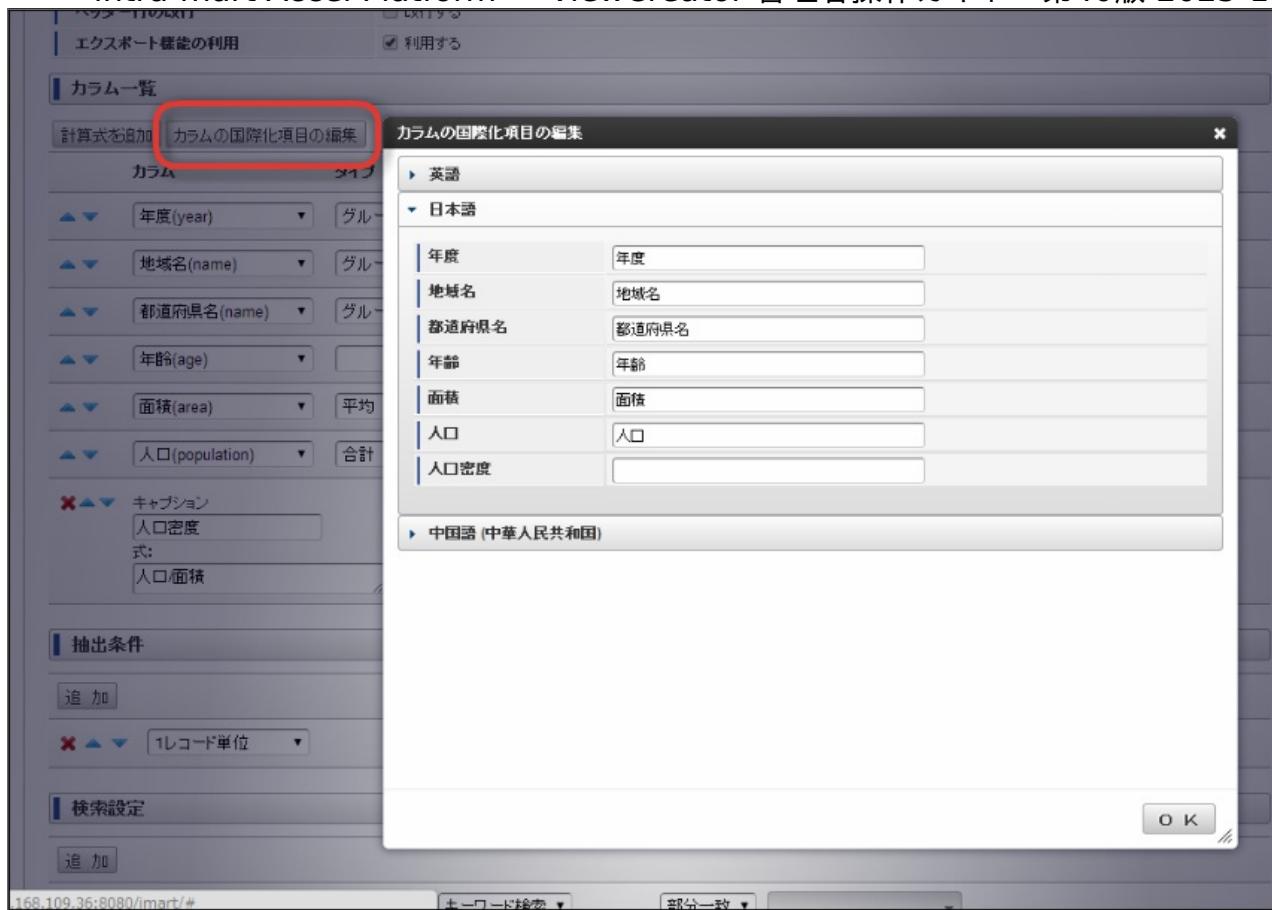
ヘッダとフッタはリッチテキストが利用可能です。

コラム

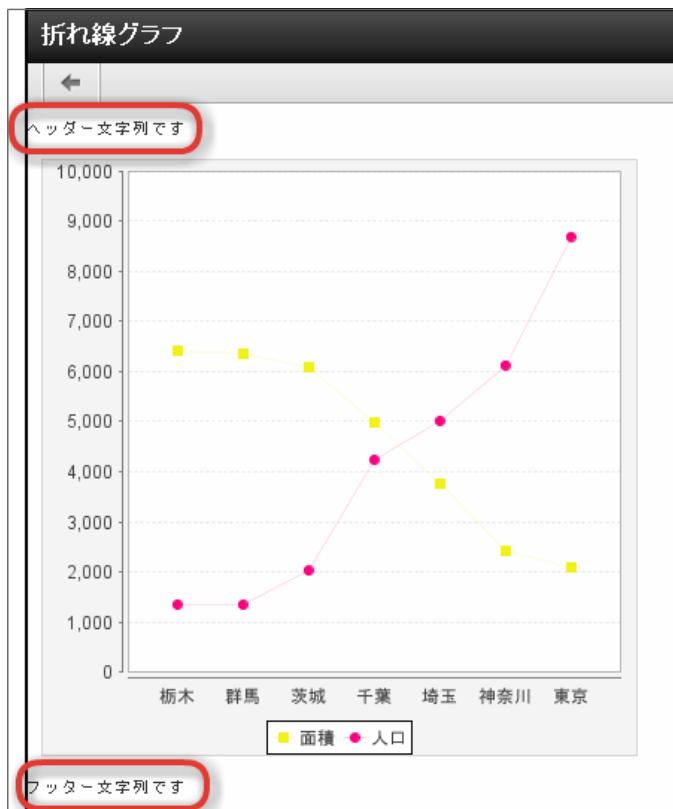
リッチテキストは2014 Spring(Granada)から利用可能です。

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。

CSV出力したファイルについても同様に、カラムヘッダには「カラムの国際化項目の編集」で設定した値が出力されます。

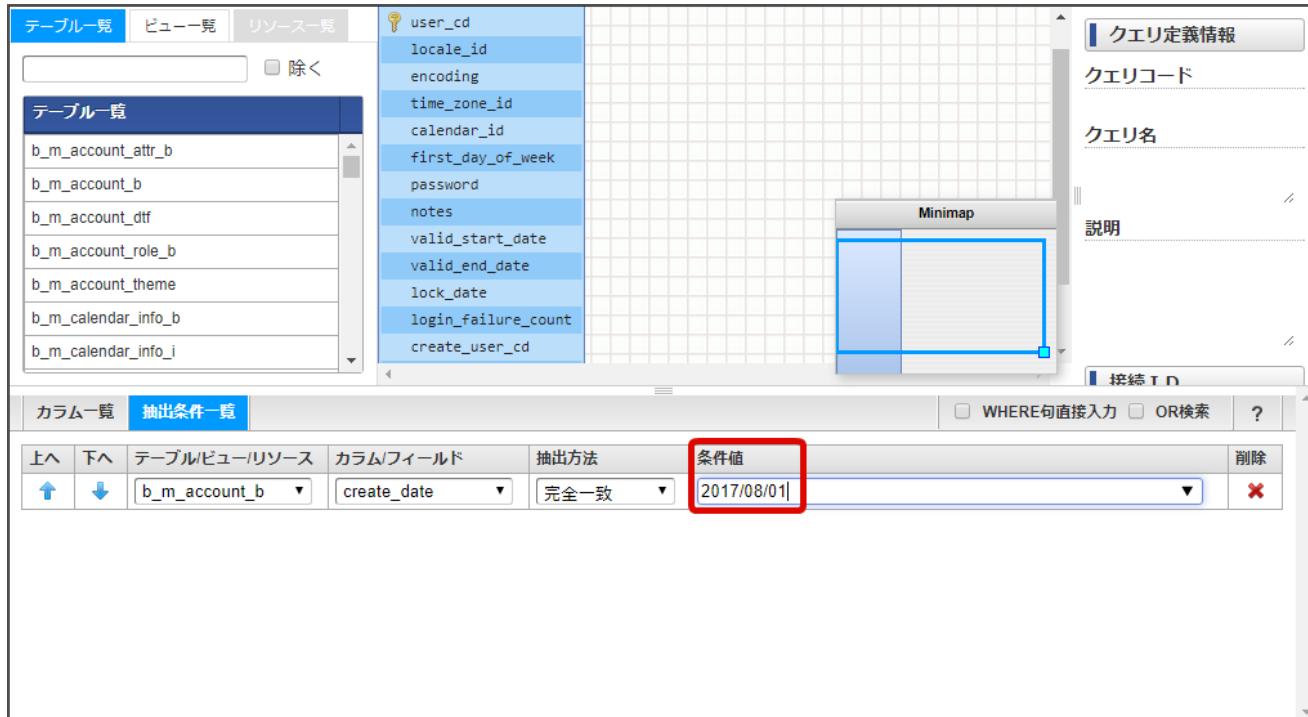


■ 表示例



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、そのことをユーザー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

- 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

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. エクスポートデータは、intra-mart Accel Platformのバージョン間におけるサポートは以下の通りです。
 - 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。
2. 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。
3. 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。
4. 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）
5. 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。

注意

異なるデータベース製品・データベースバージョン間でインポート・エクスポートを実施する場合は、データベース製品・データベースバージョン間のSQL記述の違いにより正常に動作しない可能性があります。

インポート先のデータベース製品・データベースバージョンに合わせて、クエリ定義の以下の記述を見直してください。

- SQLビルダ
- Where句直接入力
- 抽出方法-直接入力による条件値

インポート画面を確認する

「サイトマップ」→「ViewCreator - インポート」を選択します。

ViewCreator インポート

エクスポート

インポート設定

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

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

インポート種別 新規 更新

インポートファイル 選択されていません

インポート実行

インポート種別を選択する

インポート種別は「新規」または「更新」の2種類が存在します。

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

インポートを実行する

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

ViewCreator インポート

エクスポート

インポート設定

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

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

インポート種別 新規 更新

インポートファイル viewcreator-data.zip

インポート実行

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

ViewCreator インポート結果

インポートに成功しました。

総件数	新規登録件数	スキップ件数
27	27	0

ルーティングカテゴリ

総件数	新規登録件数	スキップ件数
1	1	0

ルーティング定義

総件数	新規登録件数	スキップ件数
2	2	0

クエリ定義

総件数	新規登録件数	スキップ件数
5	5	0

データ参照定義

総件数	新規登録件数	スキップ件数
19	19	0

[インポート画面に戻る](#)

スキップ件数が1件以上の場合

ViewCreator インポート結果

インポートに成功しました。

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

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

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

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

インポート画面に戻る

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。

The screenshot shows the 'ViewCreator インポート結果' (Import Result) screen. A green banner at the top right says 'インポートに成功しました。' (Import succeeded). Below it, there are four sections with summary tables:

- インポート結果**: Shows 27 total items, 16 new registrations, and 11 updates.
- ルーティングカテゴリ**: Shows 1 total item, 0 new registrations, and 1 update.
- ルーティング定義**: Shows 2 total items, 1 new registration, and 1 update.
- クエリ定義**: Shows 5 total items, 2 new registrations, and 3 updates.
- データ参照定義**: Shows 19 total items, 13 new registrations, and 6 updates.

A button at the bottom right says 'インポート画面に戻る' (Return to Import Screen).

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。

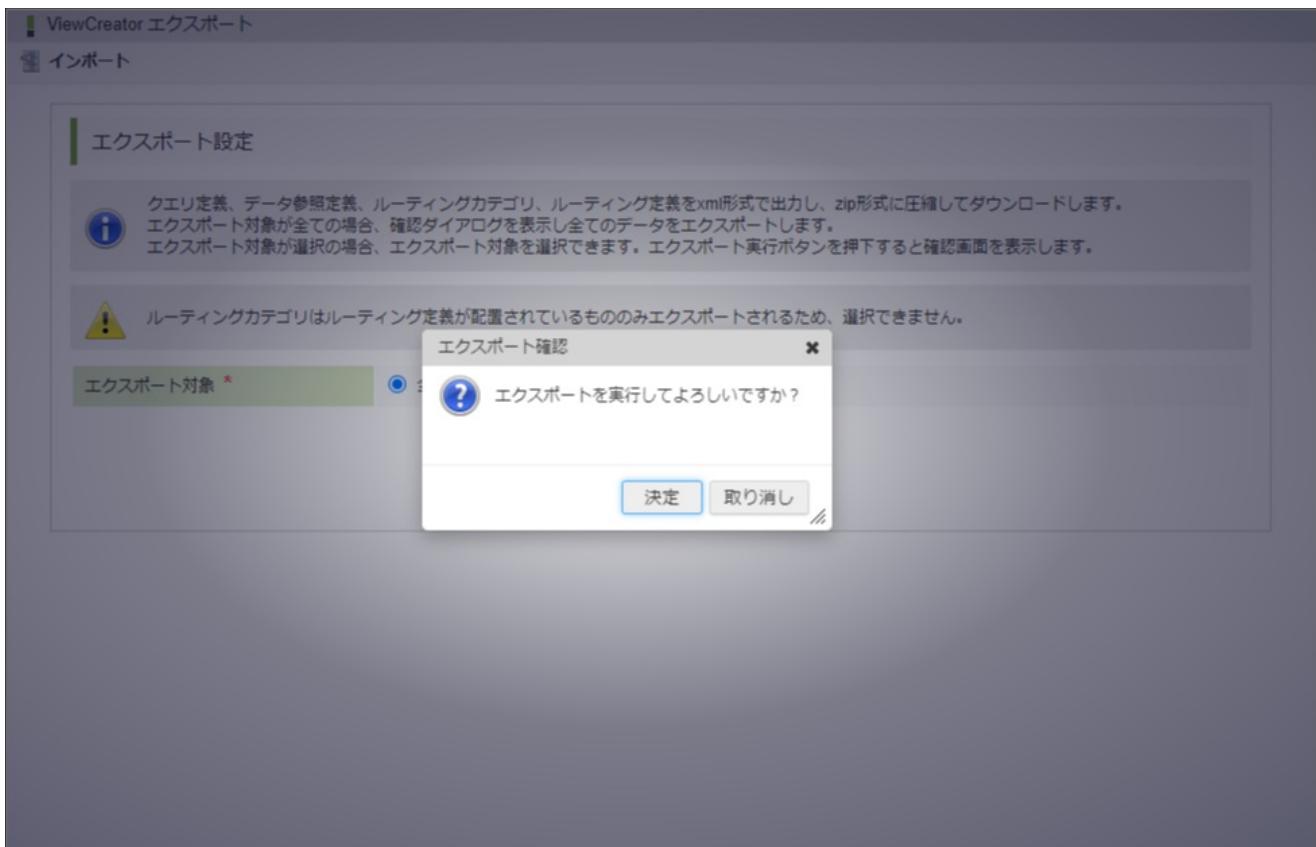
エクスポートを実行する

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。





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のみエクスポートされるため、選択できません。

エクスポート対象 *

 全て 選択

ルーティング定義

選択済み件数 : 0 件

ルーティング名検索

<input type="checkbox"/>	参照	カテゴリID	カテゴリ名	ルーティングID	ルーティング名	クエリコード	クエリ名
<input type="checkbox"/>		sample_category	サンプルカテゴリ	8g6ty5f4l32w8	エラーログ	5ib8yonru57ws	system_log
<input type="checkbox"/>		sample_category	サンプルカテゴリ	8g6ty2ma032w7	システムログ	5ib8yonru57ws	system_log

クエリ定義

選択済み件数 : 0 件

クエリ名検索

<input type="checkbox"/>	参照	クエリコード	クエリ名	説明
<input type="checkbox"/>		5i7o4o3kli1pj	日本のデータ	
<input type="checkbox"/>		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	
<input type="checkbox"/>		5ib8yonru57ws	system_log	
<input type="checkbox"/>		5ib8ony9uhzy	transition_log	
<input type="checkbox"/>		8fhil6zaswe6qchu	日本のデータ (データ結合)	

データ参照定義

選択済み件数 : 0 件

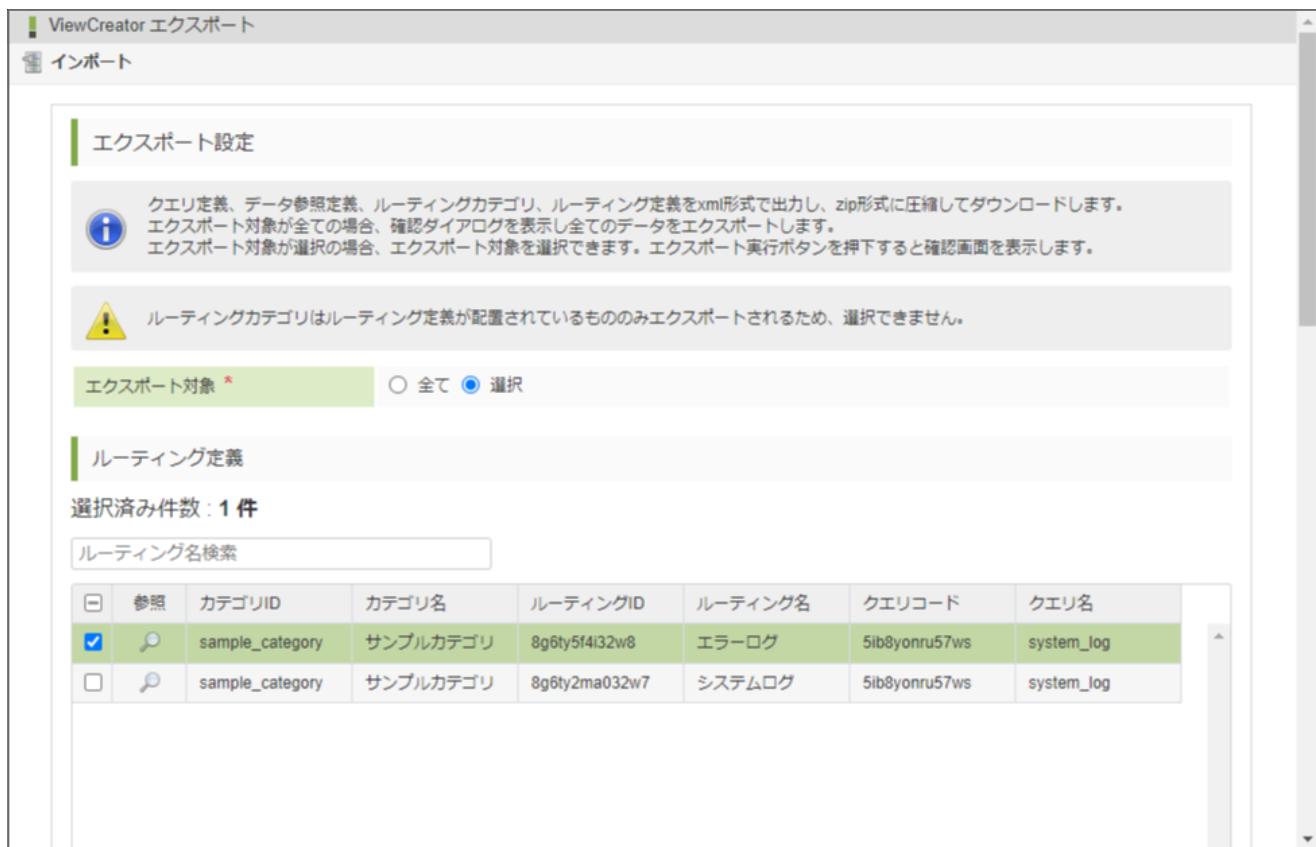
データ参照名検索

<input type="checkbox"/>	参照	データ参照コード	データ参照名	集計パターン	説明
<input type="checkbox"/>		5ib8yooamstr9v	[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	
<input type="checkbox"/>		5i7urzmesvqqip4	2000年度版・地域 - 都道府県 サマリ集計	サマリ	
<input type="checkbox"/>		5i7urzmibc9svp4	2000年度版・地域 - 都道府県 サマリ集計/面積順	サマリ	
<input type="checkbox"/>		8fhjsbw2gwvy4hu	2010年度版・地域 - 都道府県 サマリ集計	サマリ	
<input type="checkbox"/>		8fhjsvr2fwzi5hu	2010年度版・地域 - 都道府県 サマリ集計/面積順	サマリ	

1 ◀◀ 1ページ中 1 ページ目 ▶▶ ▶ 50 5 件中 1 - 19 を表示

エクスポート実行

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。

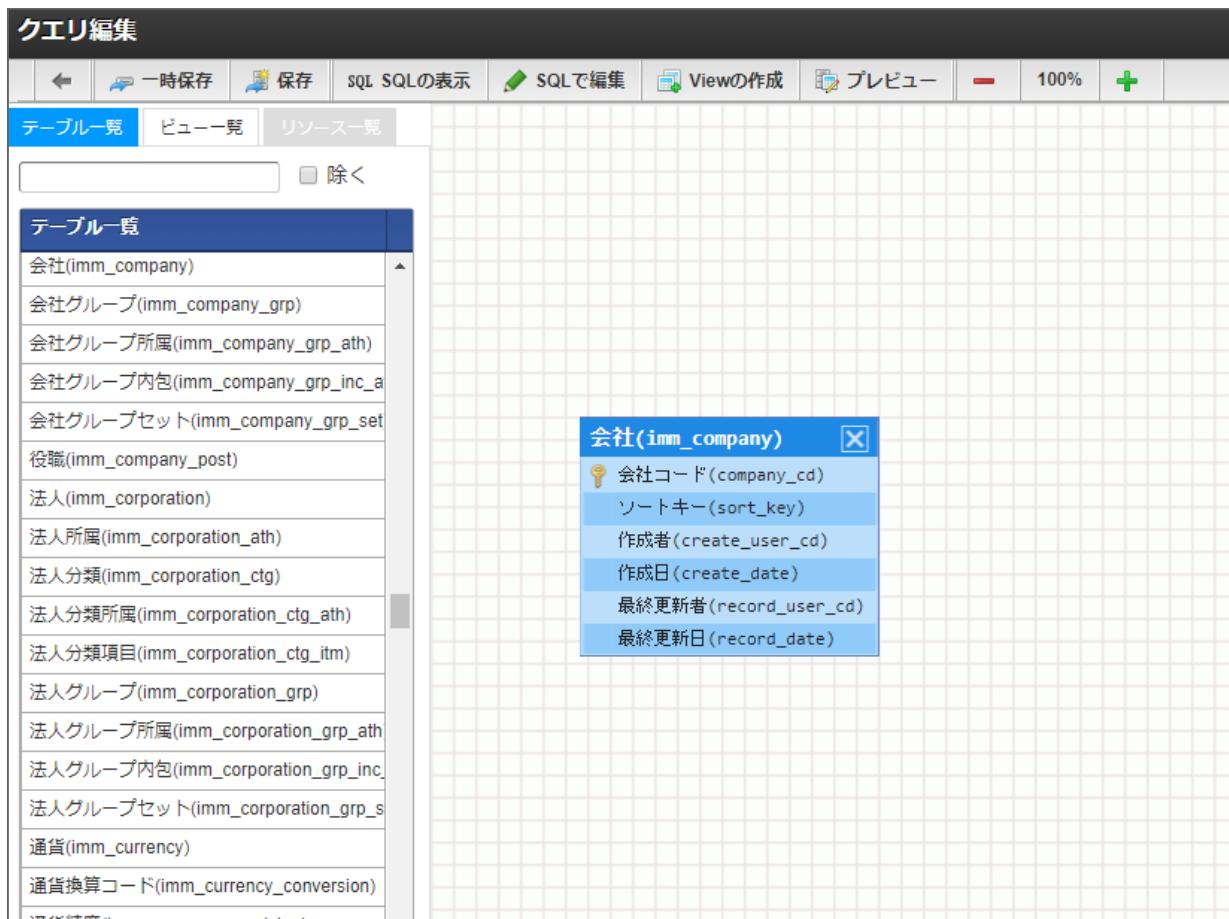


3. 「ViewCreator エクスポート対象確認」画面が表示されるので、エクスポート対象を確認し「決定」をクリックします。



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。



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。

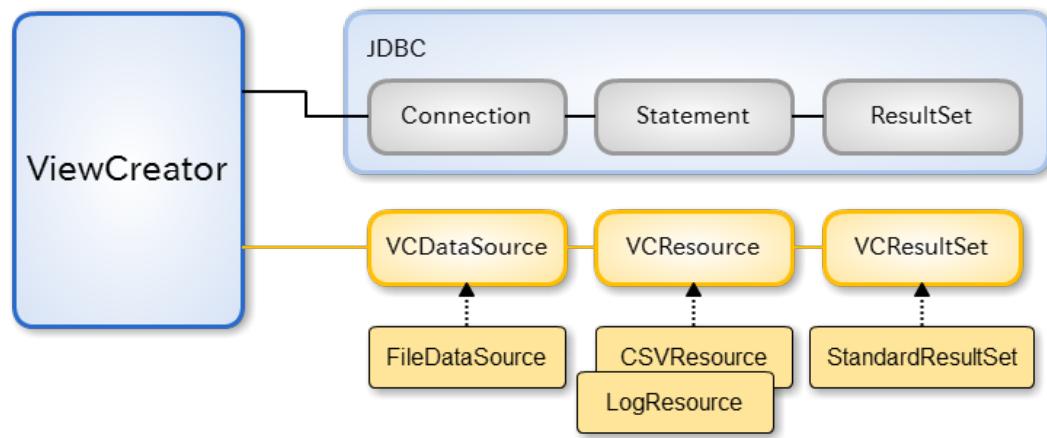
外部データソースの利用

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

目次

- [インターフェース一覧](#)
- [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](#)

インターフェース一覧



- **VCDatasource** インタフェース

このインターフェースの実装は、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

インター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。

ファイルリソース管理

リソース名	リソースパス	テーブルへ反映
japadata	c:/tmp/flatlist.csv	
system_logs	c:/tmp/log/platform/system*.log	
yuubin	c:/tmp/yuubin/ken_all.csv	

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。

ファイルリソース管理

新規作成 削除

リソース名

japadata
system_log
system_logs
transition_log
yuubin

ファイルリソース

リソース名 *: japadata
エンコーディング *: UTF-8
リソースパス *: c:/tmp/flatlist.csv
リソースクラス名:
データベースとの連携: テーブルを作成してデータ連動する
セパレータ:
日付フォーマット: [YYYY-MM-DD HH:mm:ss.SSS]
最大グループ数:
フィールド一覧 *

フィールド名	データ型
year	数値
region	文字列
prefecture	文字列
area	数値
population	数値

登録 プレビュー

リソースの設定項目

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

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



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 output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。

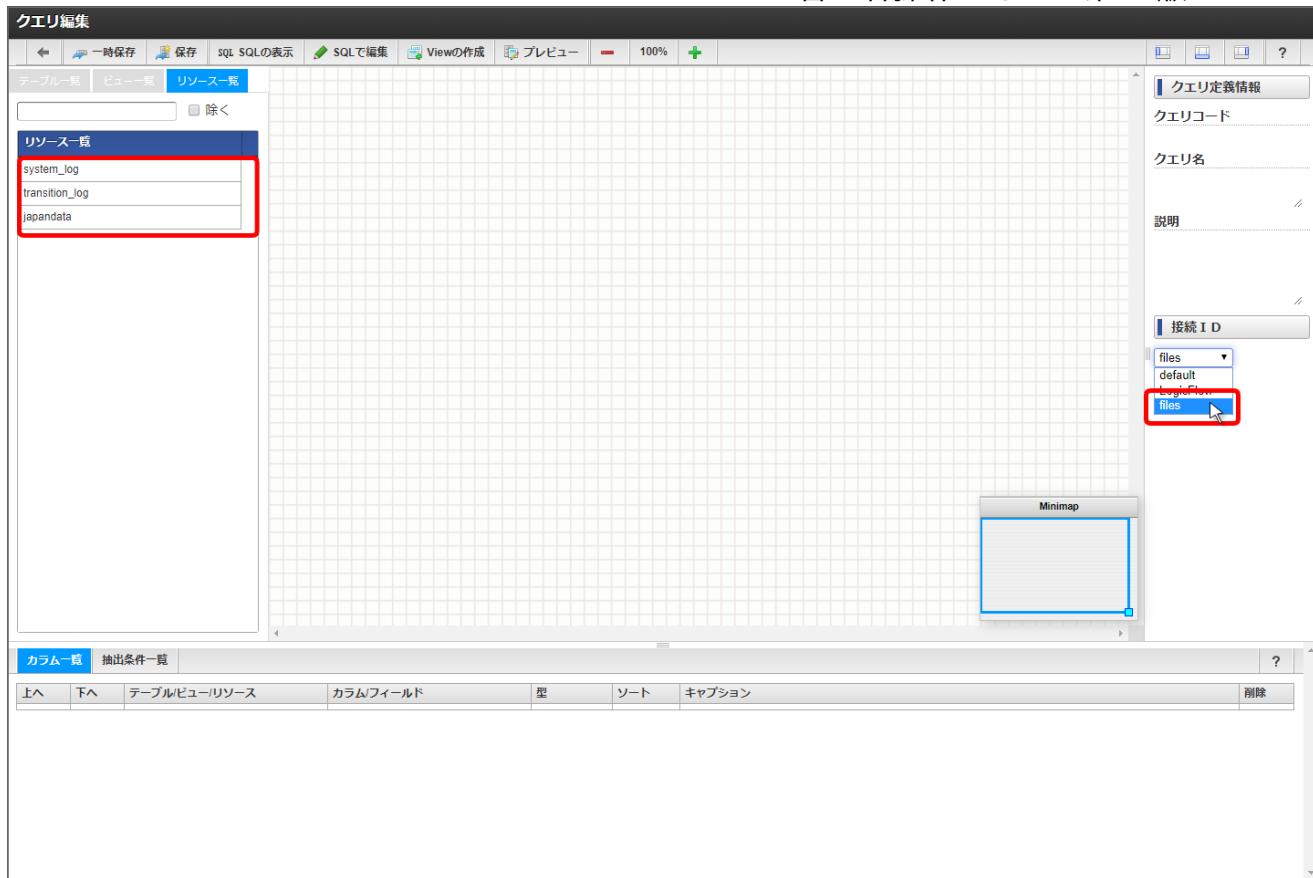
従って、これらの標準クラスを使用して作成されたリスト集計ではソートに関する全ての機能を利用できません。

「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。

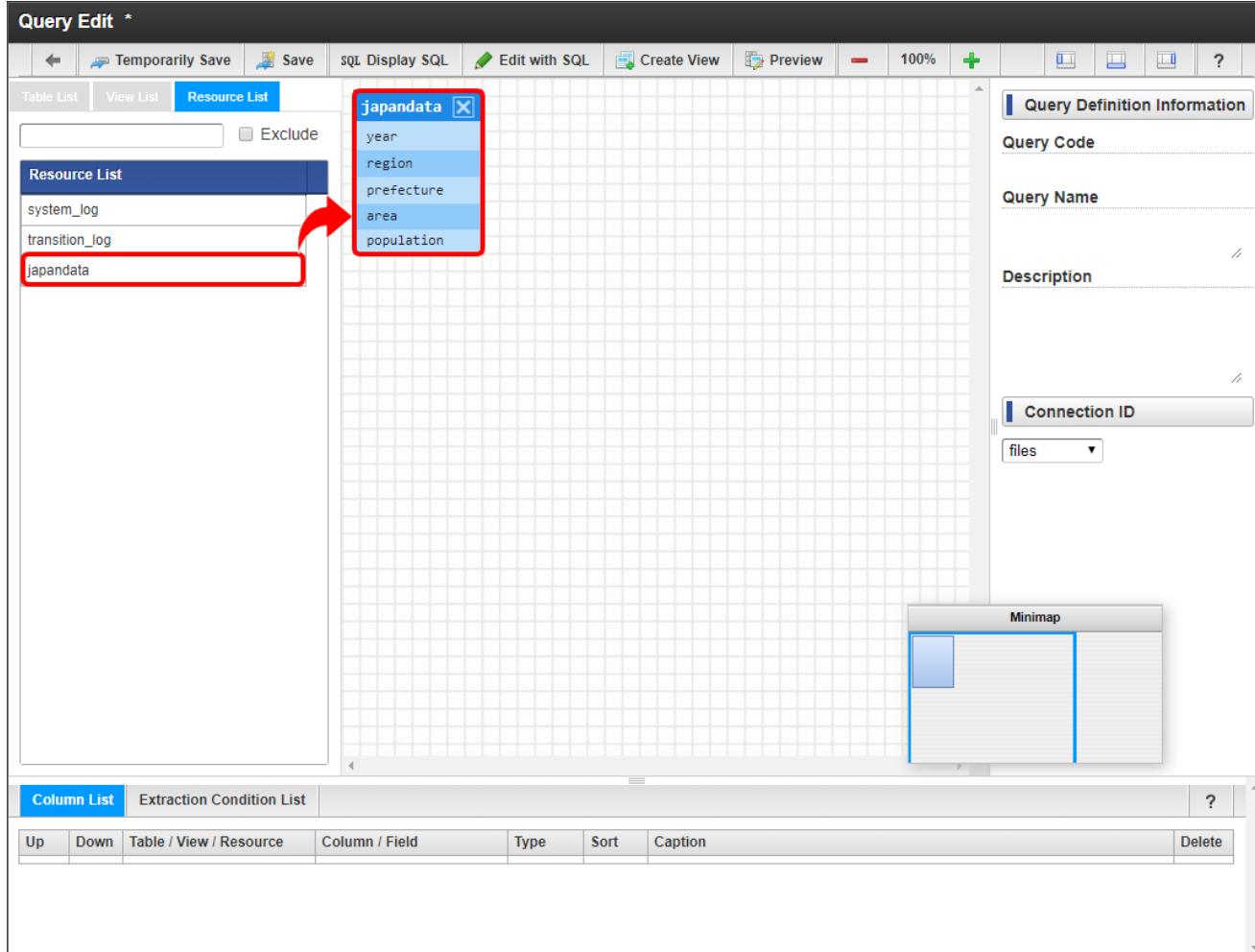


コラム

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。

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

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。

The screenshot shows the 'File Resource Management' interface. It lists three resources: 'japadata' (resource path: c:/tmp/flatlist.csv), 'system_logs' (resource path: c:/tmp/pilog/platform/system*.log), and 'yuubin' (resource path: c:/tmp/yuubin/ken_all.csv). The 'Table Reflect' column contains icons for each resource. The icon for 'japadata' is highlighted with a red box.

テーブルはテナントデータベース上に作成されます。

The screenshot shows the 'Query Editor' interface. On the left, a list of tables includes 'imvc_rc_japadata', which is highlighted with a red box. In the center, a table named 'imvc_rc_japadata' is shown with columns: f_year, f_region, f_prefecture, f_area, and f_population. The right side of the interface shows query parameters and a preview area.

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。

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。

i 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

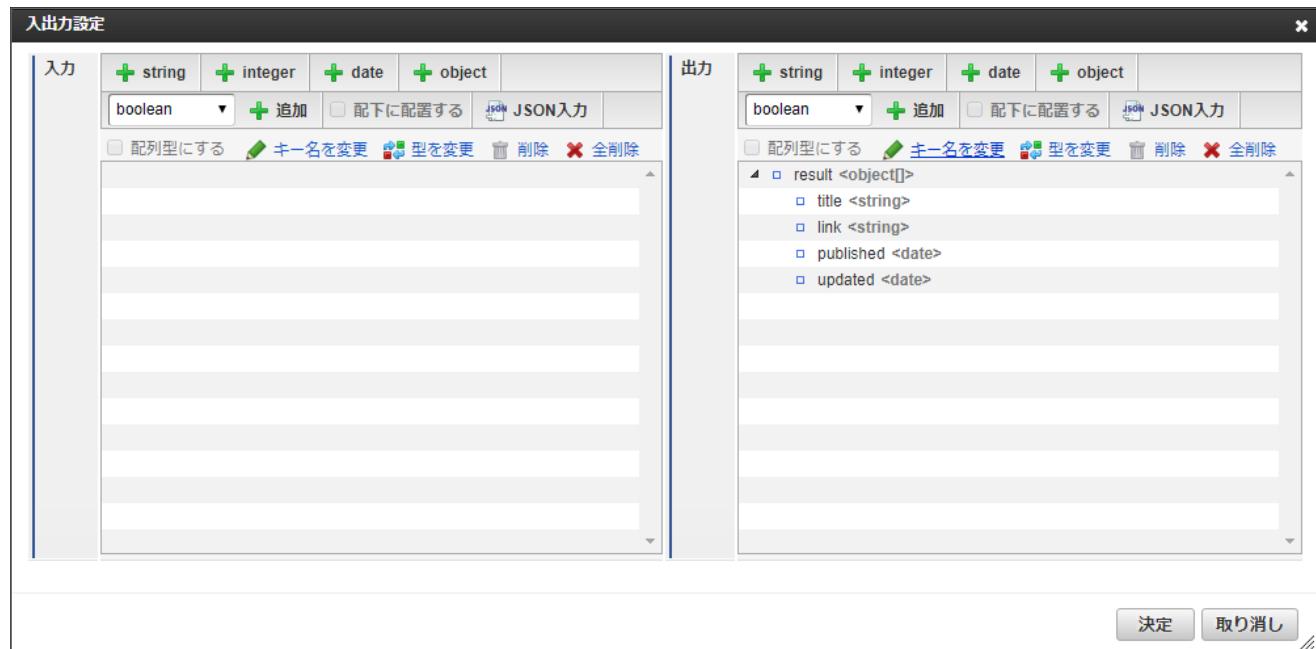
%コンテキストパス%/WEB-INF/conf/viewcreator-config.xml の<data-source-list>タグに以下の設定を追加します。

```
<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。

The screenshot shows a table titled 'Logic Flow Management' with the following data:

ViewCreatorで	フロー定義ID	フロー定義名	フローカテゴリ	プレビュー
<input checked="" type="checkbox"/>	sample-accounts	List of Accounts	Sample	
<input checked="" type="checkbox"/>	sample-backup-authz-data	Backup authz data files	Sample	
<input checked="" type="checkbox"/>	sample-im-topics-to-log	Read intra-mart atom feed	Sample	
<input checked="" type="checkbox"/>	sample-notice-to-imbox-for-new-user	Notice to IMBox for new user	Sample	
<input checked="" type="checkbox"/>	sample_im_topics_to_log	Read intra-mart atom feed	Sample	

図：「ロジックフロー管理」画面 - 初期表示

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

ロジックフローを検索する

1. 画面中央の「検索条件」をクリックします。

The screenshot shows the same table as above, but the 'Search Conditions' input field at the top of the table is highlighted with a red box.

2. 検索フィールドが表示されます。

ロジックフロー管理

検索条件

フロー定義ID	<input type="text"/>
フロー定義名	<input type="text"/>
フローカテゴリ	<input type="button" value="▼"/>

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

<画面項目>

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



コラム

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

1. 検索条件を入力し、「検索」をクリックします。

ロジックフロー管理

検索条件

フロー定義ID	<input type="text"/>
フロー定義名	<input type="text"/>
フローカテゴリ	<input type="button" value="▼"/>

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

2. 検索結果が表示されます。

検索条件

ViewCreatorで使用する	フロー定義ID	フロー定義名	フローカテゴリ	プレビュー
<input checked="" type="checkbox"/>	sample-accounts	List of Accounts	Sample	



コラム

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

クエリの作成

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

Query Edit

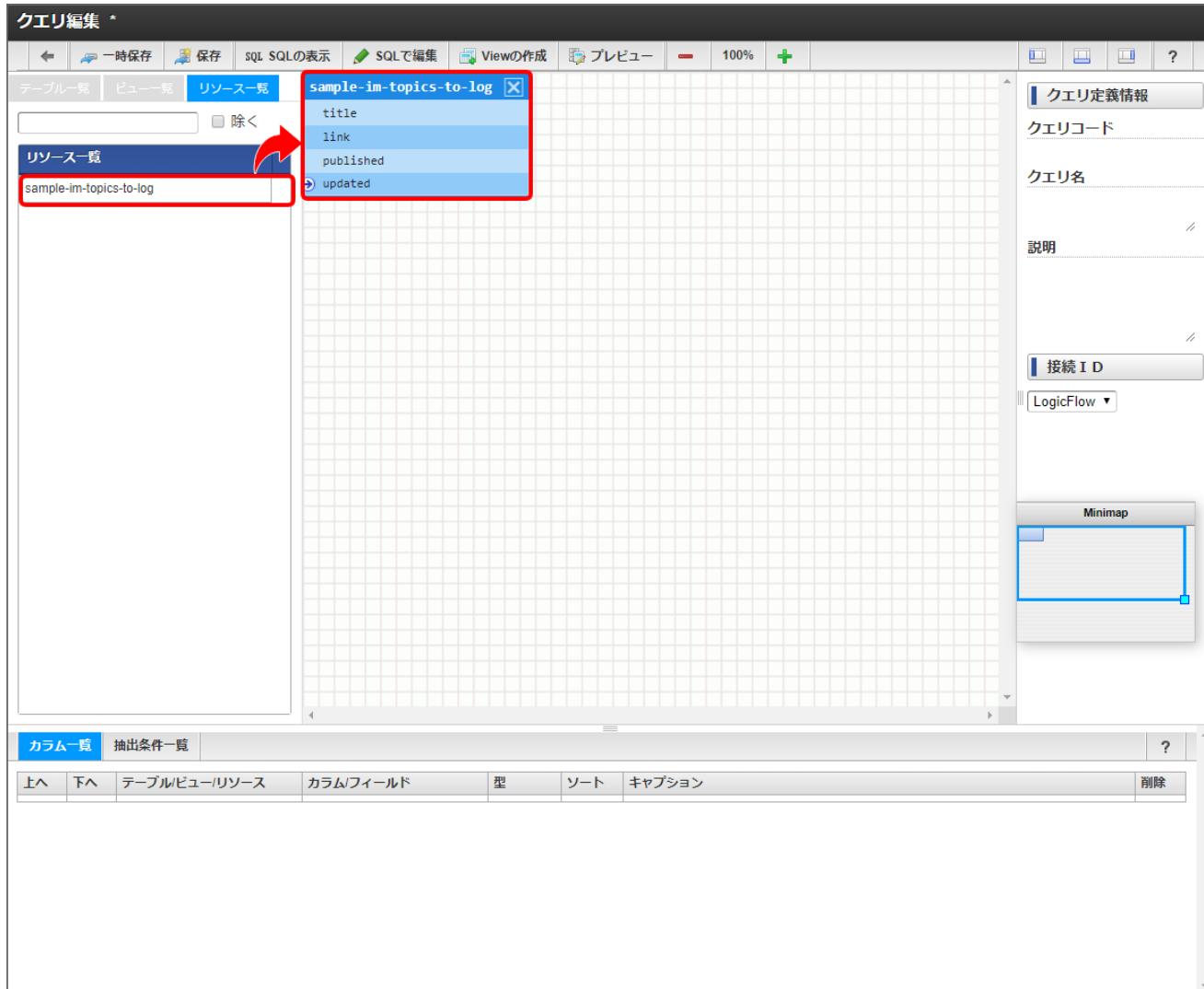
Resource List

sample-im-topics-to-log

Connection ID

LogicFlow
default
LogicFlow
files

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。

The screenshot shows the 'Column List' (カラム一覧) screen in the ViewCreator Column Editor. A red box highlights the 'Population Density' (人口密度) column, which is defined as a calculated column ('計算式を追加') with the formula '人口 / 面積' (Population / Area). The formula field contains the expression '人口/面積'.

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

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

The screenshot shows the '近畿のデータ - 1995 リスト' (Kinki Data - 1995 List) view. A red box highlights the '人口密度' (Population Density) column, which displays the calculated values for each region. The data table includes columns for Year, Region, Prefecture, Average Area, Total Population, and Population Density.

年度	地域名	都道府県名	面積(平均)	人口(合計)	人口密度
1995	近畿	三重	5,761	304	0.05
1995	近畿	滋賀	3,855	231	0.06
1995	近畿	京都	4,613	390	0.08
1995	近畿	大阪	1,894	1,321	0.7
1995	近畿	兵庫	8,394	880	0.1
1995	近畿	奈良	3,691	232	0.06
1995	近畿	和歌山	4,726	176	0.04
1995	近畿	三重	5,761	1,240	0.22
1995	近畿	滋賀	3,855	874	0.23
1995	近畿	京都	4,613	1,842	0.4
1995	近畿	大阪	1,894	6,412	3.39
1995	近畿	兵庫	8,394	3,756	0.45
1995	近畿	奈良	3,691	999	0.27
1995	近畿	和歌山	4,726	709	0.15
1995	近畿	三重	5,761	297	0.05

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

計算式カラムの計算結果を別の計算式カラムで使用することもできます。

カラム一覧

計算式を追加 カラムの国際化項目の編集

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

実行結果

← 近畿のデータ - 1995 リスト

詳細検索 CSV出力 帳票出力

年度	地域名	都道府県名	面積(平均)	人口(合計)	人口密度	人口密度 + 2
1995	近畿	三重	5,761	304	0.05	2.05
1995	近畿	滋賀	3,855	231	0.06	2.06
1995	近畿	京都	4,613	390	0.08	2.08
1995	近畿	大阪	1,894	1,321	0.7	2.7
1995	近畿	兵庫	8,394	880	0.1	2.1
1995	近畿	奈良	3,691	232	0.06	2.06
1995	近畿	和歌山	4,726	176	0.04	2.04
1995	近畿	三重	5,761	1,240	0.22	2.22
1995	近畿	滋賀	3,855	874	0.23	2.23



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。

キャプション
人口密度
式:
人口 / 面積

キャプション
人口密度 + 2
式:
人口密度 + 2

キャプション
平均との差(人口)
式:
人口 - AVG(\$人口)

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。

i 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

年度	地域名	都道府県名	面積(平均)	人口(合計)	人口密度	人口密度 + 2	平均との差(人口)
1995	近畿	三重	5,761	304	0.05	2.05	-537
1995	近畿	滋賀	3,855	231	0.06	2.06	-610
1995	近畿	京都	4,613	390	0.08	2.08	-451
1995	近畿	大阪	1,894	1,321	0.7	2.7	480
1995	近畿	兵庫	8,394	880	0.1	2.1	39
1995	近畿	奈良	3,691	232	0.06	2.06	-609
1995	近畿	和歌山	4,726	176	0.04	2.04	-665
1995	近畿	三重	5,761	1,240	0.22	2.22	399
1995	近畿	滋賀	3,855	874	0.23	2.23	33

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

クエリで定義された型 計算式で利用した場合の型

文字列	文字列型
数値	数値型
日付	日付型
タイムスタンプ	日付型
真偽値	真偽値型
バイナリ	—
—	配列型

式入力ダイアログの利用

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

関数アイコンをクリックします。



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。

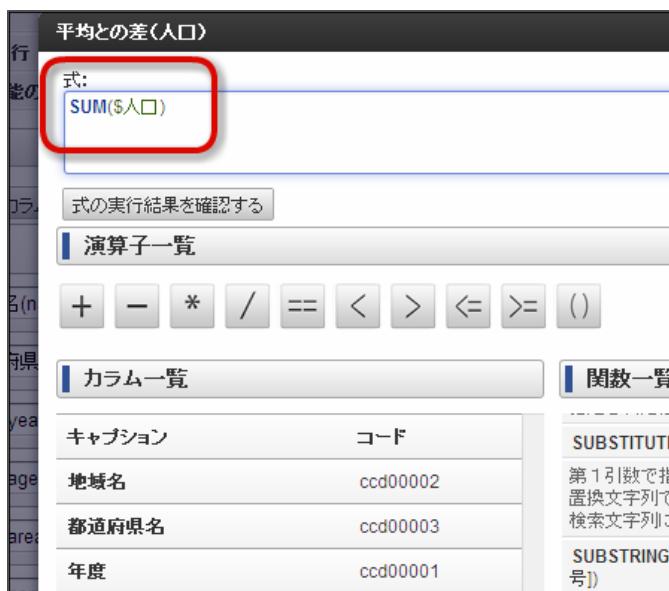


SUM関数をクリックしてみます。



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。



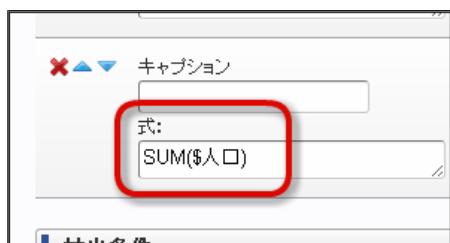
注意

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

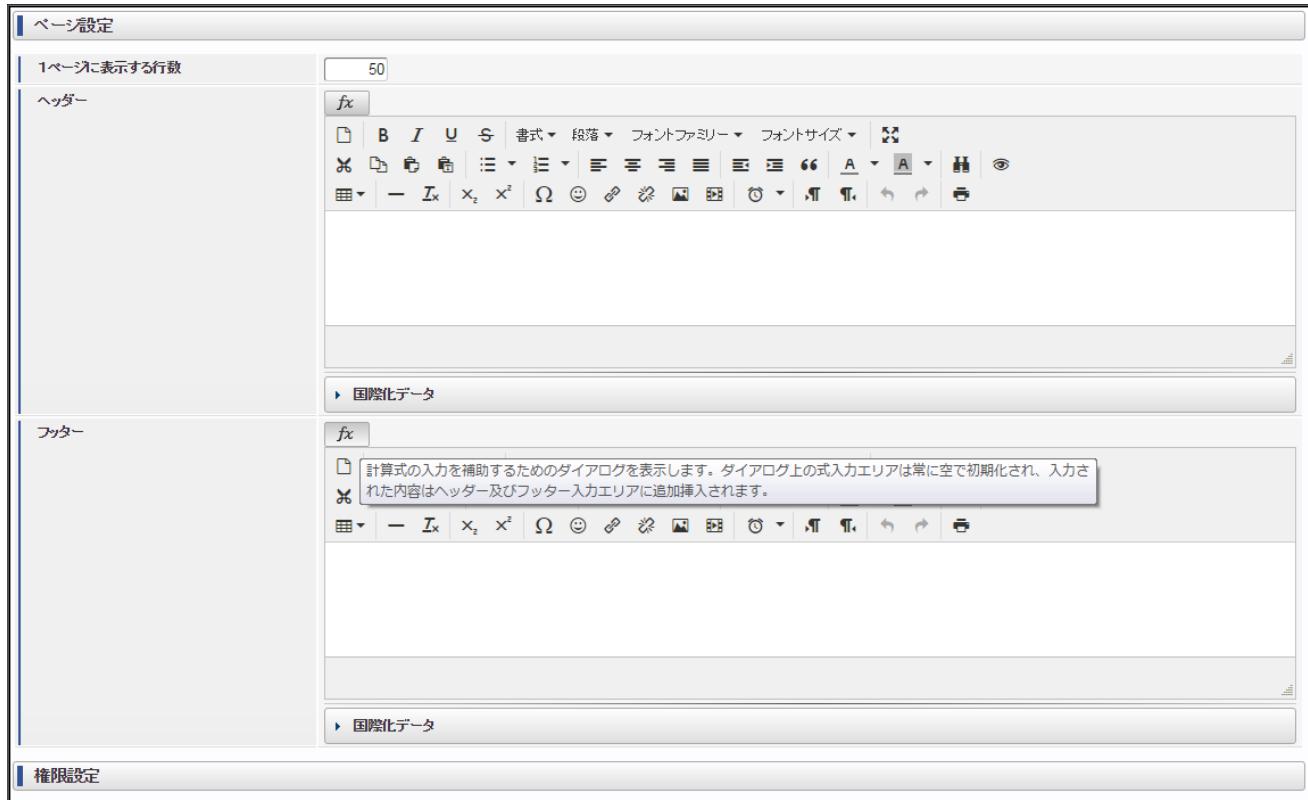
カラムの指定にはキャプションの代わりに内部的に生成されたコードを使用することもできます。キャプションを利用した場合、そのキャプションが変更されると式が実行できなくなってしまいます。コードを利用するとキャプションの変更に影響されることなく式の実行が可能ですが、但し式の可読性が落ちるため、適宜使い分けてご利用ください。

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。



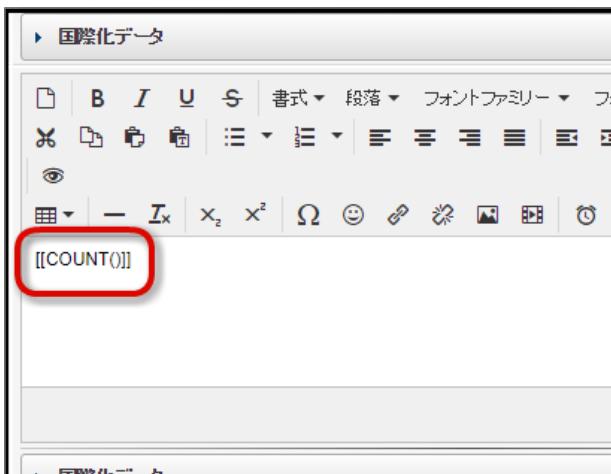
ヘッダとフッタの計算式設定

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

The screenshot shows the formula editor interface with two rows of calculated fields:

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

Below the editor is a status bar with the text "バス:" and a navigation menu with "国際化データ".

- 表示例

The screenshot shows a regional grouping list view titled "地域 - 都道府県 - 年度 グループ化リスト(人口)". The main table displays population data for various regions. A red box highlights the footer summary row, which contains the total population and the date "令和7年03月17日".

地域名	都道府県名	年度	年齢	人口(合計)
中部	-	-	-	141,850 ★
四国	-	-	-	28,852 ★
中国	-	-	-	52,922 ★
関東	-	-	-	252,118 ★
北海道	-	-	-	38,752
九州	-	-	-	98,906 ★
東北	-	-	-	66,938 ★
近畿	-	-	-	149,646 ★
				829,984
< 1 > 1 - 8 / 8				
?	レコード総数	987 件		
□	日付	令和7年03月17日		



注意

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

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

利用可能な関数一覧

ARRAY (定数値...)



コラム

ARRAY関数は2014 Summer(Honoka)から利用可能です。

- 戻り値の型：配列型

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

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

AVG (\$カラム名)

- 戻り値の型：数値型

指定されたカラムの全レコードの平均値を返します。

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



コラム

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



注意

COLUMNVALUES関数はRDBのデータソースのみ使用可能です。

- 戻り値の型：配列型

データ参照から評価式に合致するレコードの取得対象カラムを配列で取得します。

データ参照コードが省略された場合はカレントのデータ参照から取得します。

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

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

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

CONCATENATE (定数値...)

- 戻り値の型：文字列型

引数で指定された値を文字列連結します。

例) CONCATENATE(‘a’, ‘b’, ‘c’) -> abc

COUNT ()

- 戻り値の型：数値型

レコード総数を返します。

COUNTIF (評価式)

※この関数はRDBのデータソースのみ使用可能です

- 戻り値の型：数値型

評価式に合致するレコード数を返します。

例) COUNTIF(\$地域名 == “関東”) -> 7

DATAVIEWCODE ()

- 戻り値の型：文字列型

カレントのデータ参照コードを取得します。

DATAVIEWNAME ([ロケールを表す文字列], 【データ参照コード】)

- 戻り値の型：文字列型

データ参照名を取得します。引数が省略された場合は、それぞれカレントのロケール、カレントのデータ参照からデータを取得します。

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

DATETIMEFORMAT (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) DATETIMEFORMAT('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

- 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ーフェースを実装します

- jp.co.intra_mart.foundation.viewcreator.formula.FunctionImplementation

- void init(ViewCreatorData data);

ViewCreatorの設定データが渡されます。

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

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も変更して実行し、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。

i 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](#)

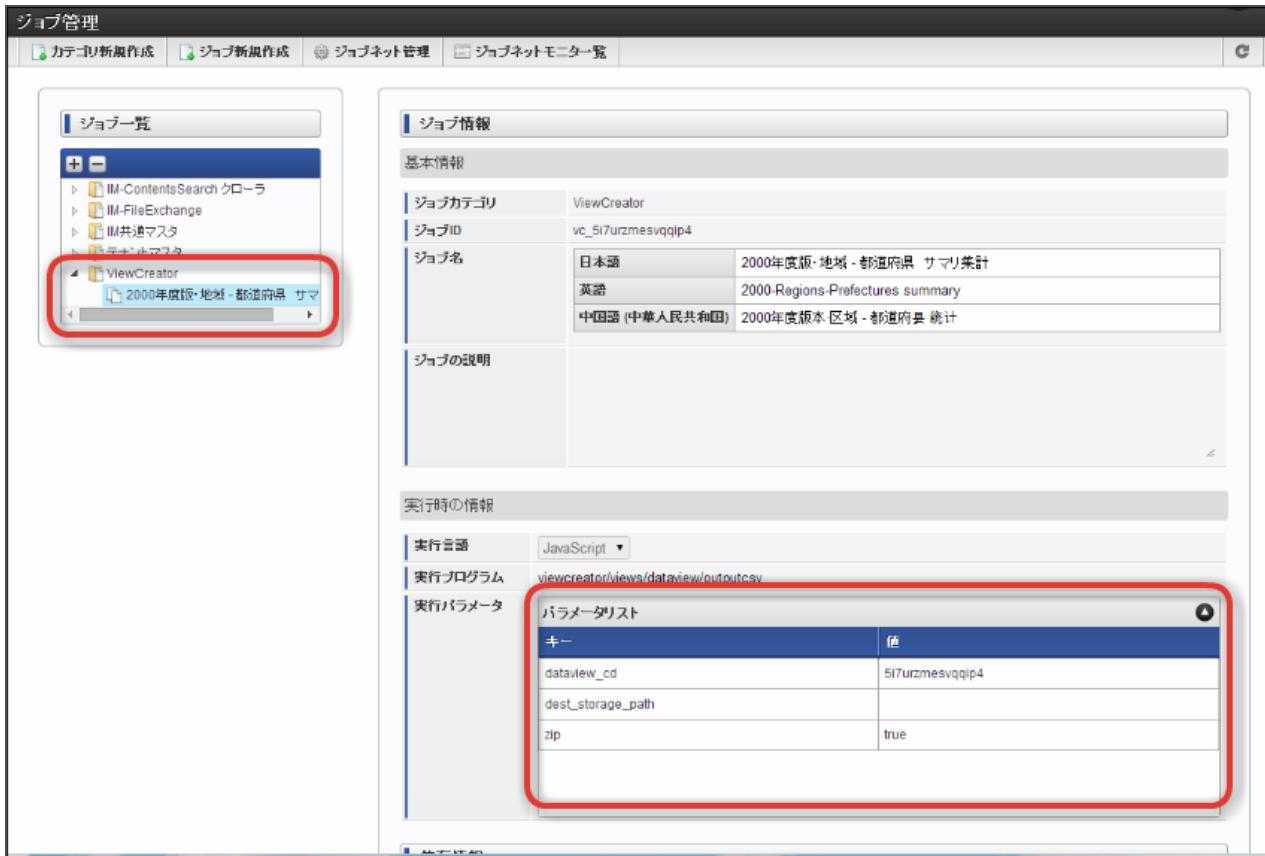
データ参照一覧から、ジョブ登録するリスト集計またはサマリ集計のデータ参照を選択し、「CSV出力ジョブの登録」を選択します。

データ参照一覧



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

サイトマップから「ジョブ管理 - ジョブ設定」を選択します。



ジョブ一覧

- IM-ContentsSearch クローラ
- IM-FileExchange
- IM共通マスター
- データ一覧マスター
- ViewCreator**
 - 2000年度版・地域 - 都道府県 サマリ

ジョブ情報

基本情報

ジョブカテゴリ	ViewCreator						
ジョブID	vc_5i7urzmesvqqip4						
ジョブ名	<table border="1"> <tr> <td>日本語</td> <td>2000年度版・地域 - 都道府県 サマリ集計</td> </tr> <tr> <td>英語</td> <td>2000-Regions-Prefectures summary</td> </tr> <tr> <td>中国語(中華人民共和国)</td> <td>2000年度版・地域 - 都道府県 統計</td> </tr> </table>	日本語	2000年度版・地域 - 都道府県 サマリ集計	英語	2000-Regions-Prefectures summary	中国語(中華人民共和国)	2000年度版・地域 - 都道府県 統計
日本語	2000年度版・地域 - 都道府県 サマリ集計						
英語	2000-Regions-Prefectures summary						
中国語(中華人民共和国)	2000年度版・地域 - 都道府県 統計						

ジョブの説明

実行時の情報

実行言語

実行プログラム

viewcreator/views/dataview/outputcsv

実行パラメータ

パラメタリスト	
キー	値
dataview_cd	5i7urzmesvqqip4
dest_storage_path	
zip	true

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]]で囲ま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。



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。

データ参照一覧

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

以下の4つのファイルがパブリックストレージに出力されます。

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

生成されたソースコードを動作させるための手順

生成されたソースコードを実際に動作させる手順を示します。

パブリックストレージに出力されたファイルのうち、jsファイルとhtmlファイルの再配置やルーティングの設定が必要です。



注意

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

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

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

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

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

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

jsファイルの編集について

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

※設定した値を、ルーティングの設定と合わせる必要があります。

1. SELF_JSSP_PATH
スクリプト開発モデルプログラムのソースディレクトリからの相対パス形式で、移動先のファイルパスをセットします。
拡張子の指定はしないでください。
2. OUTPUT_REPORT_URL
帳票出力を実行するためのURLをセットします。

3. OUTPUT_CSV_URL

CSV出力を実行するためのURLをセットします。

※クロス集計とグラフ集計の場合は不要です。

▪ 編集前jsファイルの例

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

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

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

ルーティングの設定について

下記のルーティングを作成してください。

1. 参照画面の表示用のマッピング

path属性には、データ参照の表示画面にアクセスするためのURLをセットします。

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

2. 帳票出力用のマッピング

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

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

action属性には、"outputReport"をセットします。

3. CSV出力用のマッピング ※クロス集計とグラフ集計の場合は不要です。

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

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

action属性には、"outputCSV"をセットします。

▪ 設定例

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

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

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

1. jsファイルとhtmlファイルをWEB-INF/jsspディレクトリ以下の任意のディレクトリに移動する

2. jsファイルを編集する

3. 移動先のパスに対して、スクリプト開発モデルルーティングの設定を行う

jsファイルの編集について

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

※設定した値を、ルーティングの設定と合わせる必要があります。

1. SELF_JSSP_PATH

スクリプト開発モデルプログラムのソースディレクトリからの相対パス形式で、移動先のファイルパスをセットします。

拡張子の指定はしないでください。

2. VIEW_PAGE_URL

参照画面（自分自身）を表示するURLをセットします。

3. OUTPUT_REPORT_URL

帳票出力を実行するためのURLをセットします。

4. OUTPUT_CSV_URL

CSV出力を実行するためのURLをセットします。

※クロス集計とグラフ集計の場合は不要です。

- 編集前jsファイルの例

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

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

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

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

ルーティングの設定について

下記のルーティングを作成してください。

1. 参照画面の表示用のマッピング

path属性には、jsファイルのVIEW_PAGE_URLと同じ値をセットします。
page属性には、jsファイルのSELF_JSSP_PATHと同じ値をセットします。

2. 帳票出力用のマッピング

path属性には、jsファイルのOUTPUT_REPORT_URLと同じ値をセットします。
page属性には、jsファイルのSELF_JSSP_PATHと同じ値をセットします。
action属性には、"outputReport"をセットします。

3. CSV出力用のマッピング ※クロス集計とグラフ集計の場合は不要です。

path属性には、jsファイルのOUTPUT_CSV_URLと同じ値をセットします。
page属性には、jsファイルのSELF_JSSP_PATHと同じ値をセットします。
action属性には、"outputCSV"をセットします。

- 設定例

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

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

帳票出力の設定

データ参照をPDF等の形式で出力することを可能にする設定です。

すべてのタイプのデータ参照で利用できます。



コラム

帳票出力の設定は2014 Summer(Honoka)から利用可能です。

目次

- 設定ファイルの配置場所
- 設定項目
- コマンドを利用する場合に指定するパラメータ

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。

The screenshot shows two main sections of the ViewCreator application.

Top Section (Report View):

- Title:** 年度・都道府県別人口 - クロス集計
- Buttons:** 検索, 帳票出力
- Table:** A cross-tabulation table for population by year (1990, 1995, 2000) and age group (0 - 14歳, 15 - 64歳, 65歳以上) across various Japanese regions. The table includes summary rows for北海道 and 東北, and a total row for the entire region.

		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

Bottom Section (User List View):

- Buttons:** リスト表示, 検索ボックス (検索), CSV出力, 帳票出力, カスタマイズ, フルリセット
- Table:** A list of users with columns for User Code, User Name, Telephone Number, Email Address, Sort Key, and Creation Date.

ユーザコード	ユーザ名	電話番号	メールアドレス1	ソートキー	作成日
tenant	tenant	-	-	0	2025/03/14
ueda	上田辰男	-	-	1	2012/08/08
aoyagi	青柳辰巳	-	-	2	2012/08/08
hayashi	林政義	-	-	3	2012/08/08
maruyama	円山益男	-	-	4	2012/08/08
sekine	関根千香	-	-	5	2012/08/08
terada	寺田雅彦	-	-	6	2012/08/08
yoshikawa	吉川一哉	-	-	7	2012/08/08
ohiso	大磯博文	-	-	8	2012/08/08
hagimoto	萩本順子	-	-	9	2012/08/08
ikuta	生田一哉	-	-	10	2012/08/08
katayama	片山聰	-	-	11	2012/08/08
harada	原田浩二	-	-	12	2012/08/08

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>COMMAND_NAME %INPUT_PATH% %OUTPUT_PATH%</report-create-command>
<file-suffix>pdf</file-suffix>
</report-settings>
```

i 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。

i 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、下記のインター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。

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。



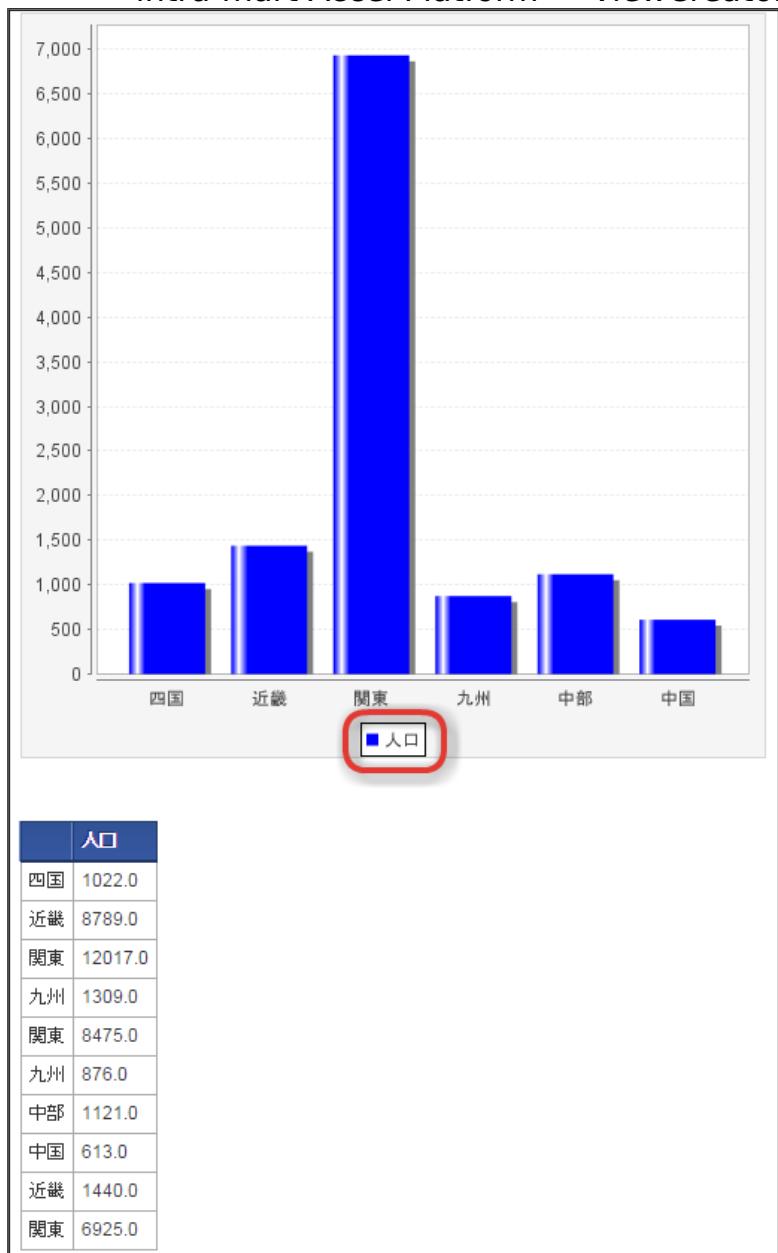
コラム

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



凡例として「人口」というデータが表示されています。

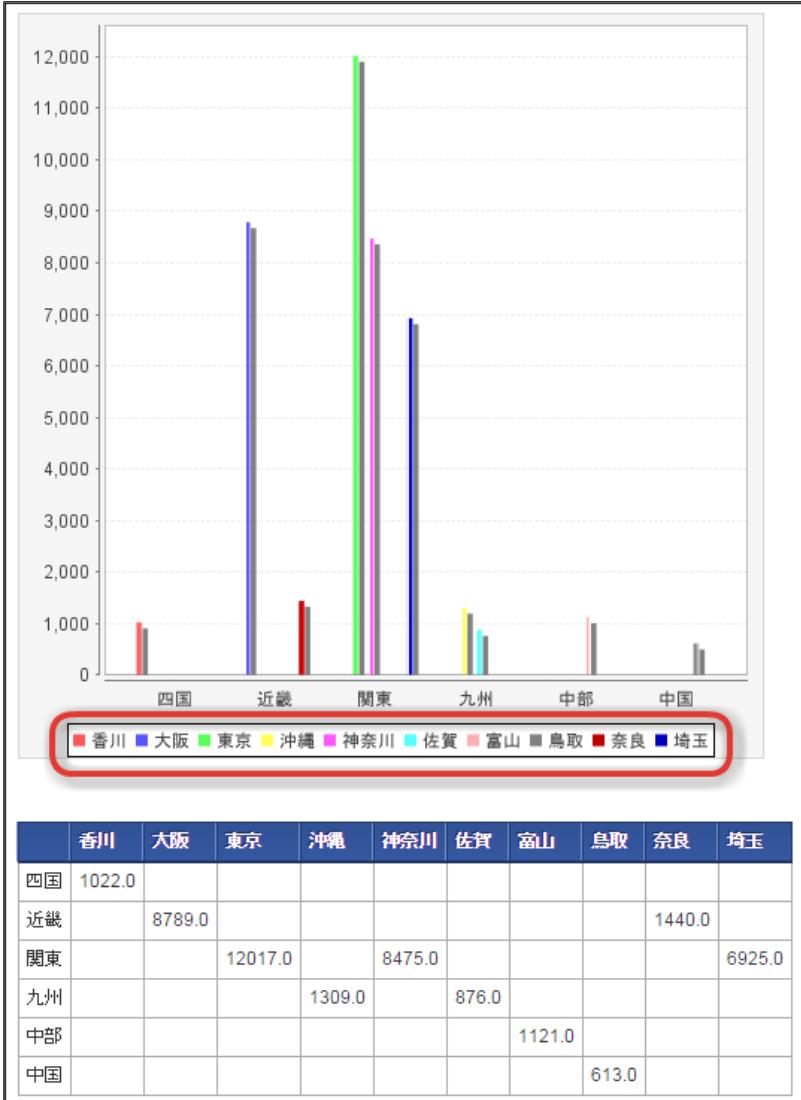
The screenshot shows the 'Column Settings' dialog box. It includes sections for 'Caption Column', 'Example Code Column', 'Example Label Column', and 'Value Column'. Below these are settings for 'Display Column' and 'Color'. The 'Value Column' section shows four items: '年度' (Year) with color #000000, '面積' (Area) with color #ff0000, '人口' (Population) with color #0000ff (highlighted with a red box), and '人口密度' (Population Density) with color #000000.

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

凡例カラムの設定を行った場合



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



i コラム

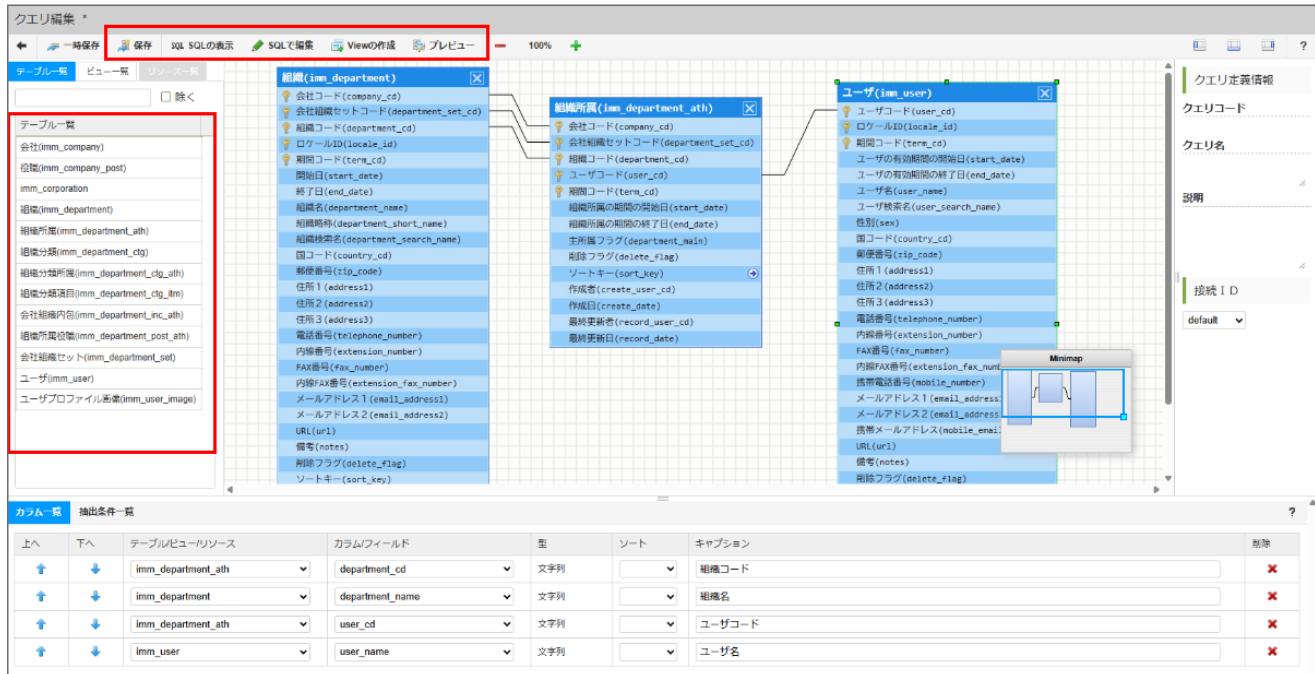
「凡例コードカラム」に設定されたカラムから取得されたデータをキーとして、同じ凡例項目に属するデータであるかどうかの判定が行われます。

画面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

クエリ編集画面でテーブルを扱うすべての機能について、テーブル単位のアクセス権により利用可能なテーブルが制限されます。具体的に制限が適用される機能は以下の通りです。

- テーブル一覧
- 保存
- SQLで編集（SQLビルダ）
- SQLビルダ アシスタント
- Viewの作成
- プレビュー



テーブル一覧には、アクセス権のあるテーブルのみが表示されます。

保存、Viewの作成、プレビューでは、アクセス許可されていないテーブルが含まれている場合、エラーが表示されます。

SQLビルダ、SQLビルダ アシスタントでは、アクセス権のあるテーブルだけが利用できます。



コラム

クエリー一覧画面では、クエリビルダ、SQLビルダどちらで作成されたクエリかによって以下の違いがあります。

- クエリビルダで作成されたクエリの場合
アクセス許可されていないテーブルが含まれているクエリは一覧に表示されません。
- SQLビルダで作成されたクエリの場合
「SQLビルダ」の認可リソースが許可されているユーザは、SQLビルダで作成されたクエリが全て一覧に表示されます。



注意

2025 Spring(Kamille) 以前では、テーブル単位のアクセス権設定は、クエリビルダで作成されたクエリにのみ適用されていました。

2025 Autumn(Lilac) 以降では、SQLビルダ等、SQLの中に含まれるテーブルに対してもアクセス権が適用されるようになりました。

目次

- 保存、Viewの作成、プレビュー
- SQLビルダ
- SQLビルダ アシスタント
- 設定方法

保存、Viewの作成、プレビュー

実行時にアクセス権のないテーブルが含まれている場合、エラーが表示されます。

Viewの作成

アクセス許可されていないテーブルが含まれています。: b_m_account_b

imm_department

- 会社コード(company_cd)
- 組織セットコード(department_set_cd)
- 組織コード(department_cd)
- ロケルID(locale_id)
- 期間コード(term_cd)
- 始日(start_date)
- 終日(end_date)
- 部門名(department_name)
- 略称(department_short_name)
- 検索名(department_search_name)
- 国コード(country_cd)
- 郵便番号(zip_code)
- 住所1(address1)
- 住所2(address2)
- 住所3(address3)
- 電話番号(telephone_number)
- 内線番号(extension_number)
- fax番号(fax_number)
- 内線FAX番号(extension_fax_number)
- メールアドレス1(email_address1)
- メールアドレス2(email_address2)
- (url)
- 備考(notes)
- 削除フラグ(delete_flag)
- トキー(sort_key)

組織所属(imm_department_ath)

- 会社コード(company_cd)
- 会社組織セットコード(department_set_cd)
- 組織コード(department_cd)
- ユーザコード(user_cd)

ユーザ(imm_user)

- ユーザコード(user_cd)
- ロケルID(locale_id)
- 期間コード(term_cd)
- ユーザの有効期間の開始日(start_date)
- ユーザの有効期間の終了日(end_date)
- ユーザ名(user_name)
- ユーザ検索名(user_search_name)
- 性別(sex)
- 国コード(country_cd)
- 郵便番号(zip_code)
- 住所1(address1)
- 住所2(address2)
- 住所3(address3)
- 電話番号(telephone_number)
- 内線番号(extension_number)
- FAX番号(fax_number)
- 内線FAX番号(extension_fax_number)
- 携帯電話番号(mobile_number)
- メールアドレス1(email_address1)
- メールアドレス2(email_address2)
- 携帯メールアドレス(mobile_email)
- URL(url)
- 備考(notes)
- 削除フラグ(delete_flag)

Viewの作成

```
CREATE OR REPLACE VIEW imm_department_ath AS
SELECT
    imm_department_ath.department_cd AS department_cd,
    imm_department.department_name AS department_name,
    imm_department_ath.user_cd AS user_cd,
    imm_user.user_name AS user_name

FROM
    imm_department
    INNER JOIN imm_department_ath
        ON imm_department.company_cd = imm_department_ath.company_cd
        AND imm_department.department_set_cd =
            imm_department_ath.department_set_cd
        AND imm_department.department_cd = imm_department_ath.department_cd
    INNER JOIN imm_user
        ON imm_department_ath.user_cd = imm_user.user_cd
    INNER JOIN b_m_account_b
        ON imm_user.user_cd = b_m_account_b.user_cd
```

実行 キャンセル

カラム/フィールド	型	ソート	キャプション
department_cd	文字列		組織コード
department_name	文字列		組織名

SQLビルダ

アクセス権のないテーブルを利用したクエリを作成しようとすると、エラーが表示されます。

クエリ編集 *

ViewCreator SQLビルダ アシスタント

アクセス許可されていないテーブルが含まれています。: b_m_account_b

```
1  SELECT
2      imm_department_ath.department_cd AS department_cd,
3      imm_department.department_name AS department_name,
4      imm_department_ath.user_cd AS user_cd,
5      imm_user.user_name AS user_name
6
7  FROM
8      imm_department
9      INNER JOIN imm_department_ath
10         ON imm_department.company_cd = imm_department_ath.company_cd
11         AND imm_department.department_set_cd = imm_department_ath.department_set_cd
12         AND imm_department.department_cd = imm_department_ath.department_cd
13     INNER JOIN imm_user
14         ON imm_department_ath.user_cd = imm_user.user_cd
15     INNER JOIN b_m_account_b
16         ON imm_user.user_cd = b_m_account_b.user_cd
```

例：ユーザの一覧を取得するSQLを作成してください。

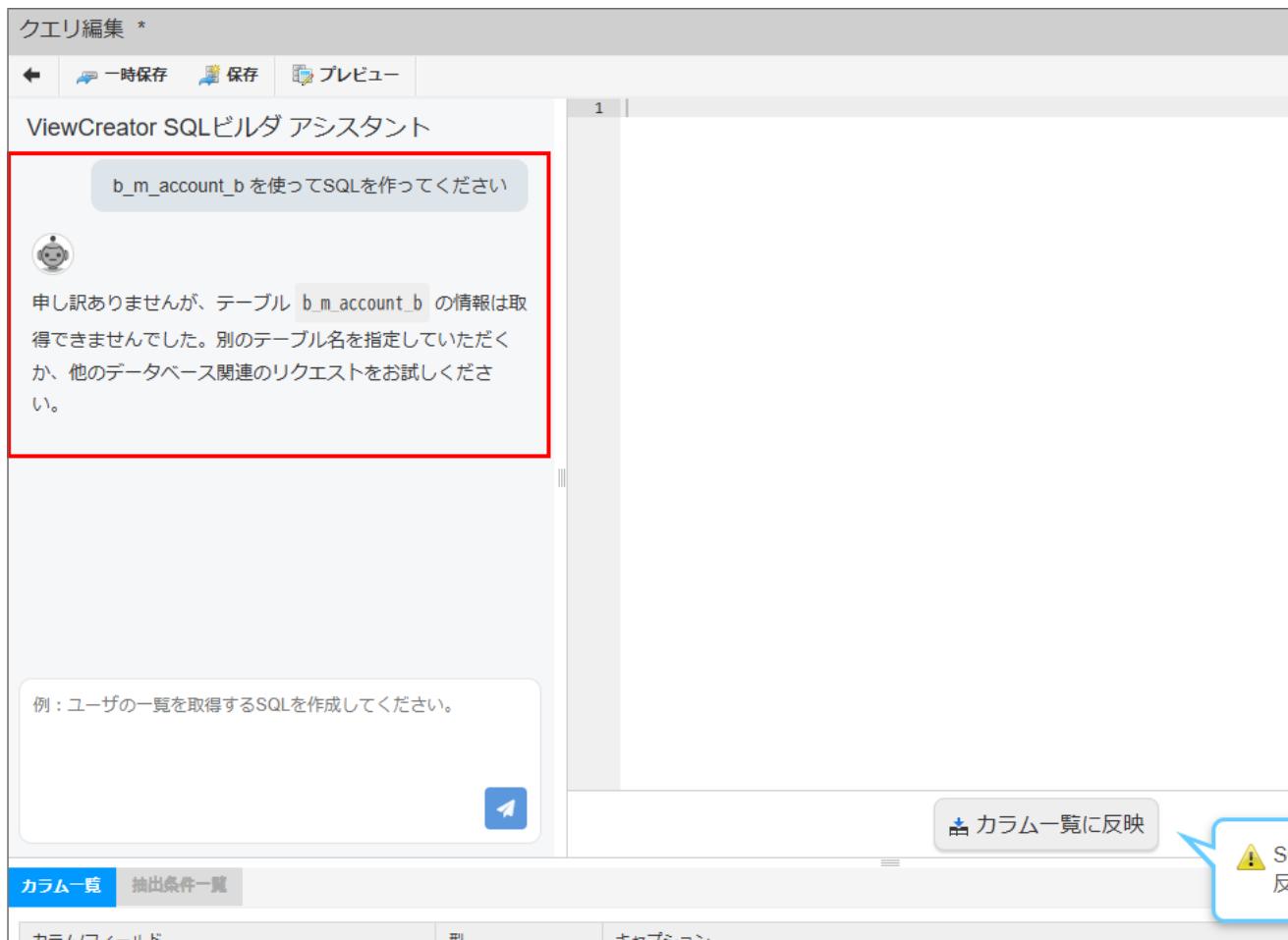
カラム一覧に反映

SQLをカラム一覧に反映してください。

カラム/フィールド	型	キャプション
department_cd	文字列	組織コード
department_name	文字列	組織名
user_cd	文字列	ユーザコード
user_name	文字列	ユーザ名

SQLビルダ アシスタントは、アクセス権のないテーブルを利用したSQLを作成できません。

また、アクセス権のないテーブルについて回答できません。



設定方法

テーブル単位のアクセス権設定で、アクセスを許可したいテーブルについて、以下のいずれかを許可します。

- 参照
- 更新
- 実行

詳細については「[メンテナンス画面のアクセス権設定](#)」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

swagger http://127.0.0.1:8080/imart/api-docs/viewcreator api_key Explore

ViewCreator

ViewCreatorに関するAPIを提供します。

Category: ルーティングカテゴリに関するAPIです。 Show/Hide | List Operations | Expand Operations

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

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

Query: クエリに関するAPIです。 Show/Hide | List Operations | Expand Operations

Route: クエリルーティングに関するAPIです。 Show/Hide | List Operations | Expand Operations

[BASE URL: /imart , API VERSION: 1.0.0]

「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。

Category: ルーティングカテゴリに関するAPIです。 Show/Hide | List Operations | Expand Operations

GET	/api/viewcreator/query/query_route/categories	ルーティングカテゴリ情報一覧を取得します。
POST	/api/viewcreator/query/query_route/categories	ルーティングカテゴリ情報を登録します。
DELETE	/api/viewcreator/query/query_route/categories/{categoryId}	ルーティングカテゴリ情報を削除します。
GET	/api/viewcreator/query/query_route/categories/{categoryId}	ルーティングカテゴリ情報を取得します。
PUT	/api/viewcreator/query/query_route/categories/{categoryId}	ルーティングカテゴリ情報を更新します。
PUT	/api/viewcreator/query/query_route/categories_sort	ルーティングカテゴリ情報のソート番号を更新します。

「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": [
      {
        "code": "string"
      }
    ]
  }
}
```

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。



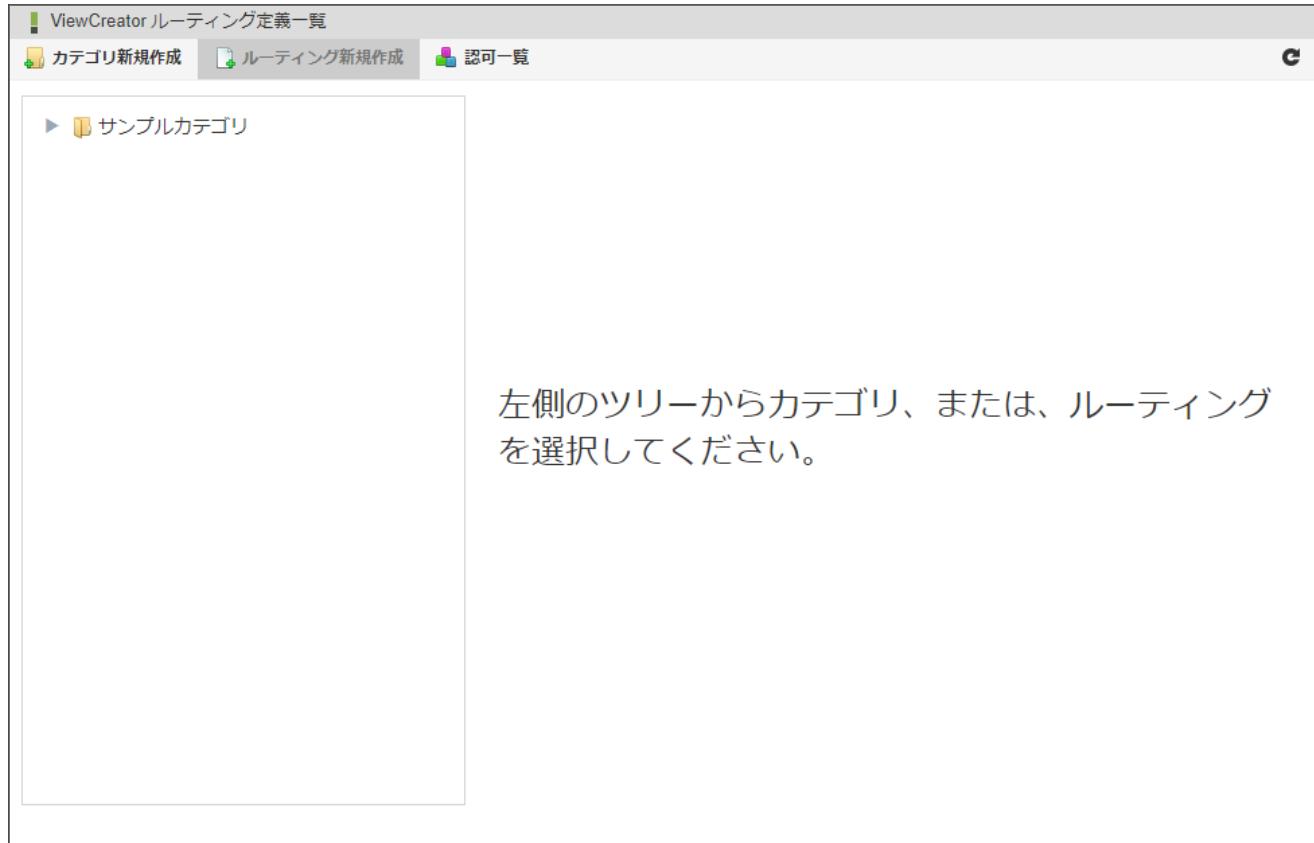
ルーティング定義作成機能は 2020 Winter(Azalea) から利用可能です。

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
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

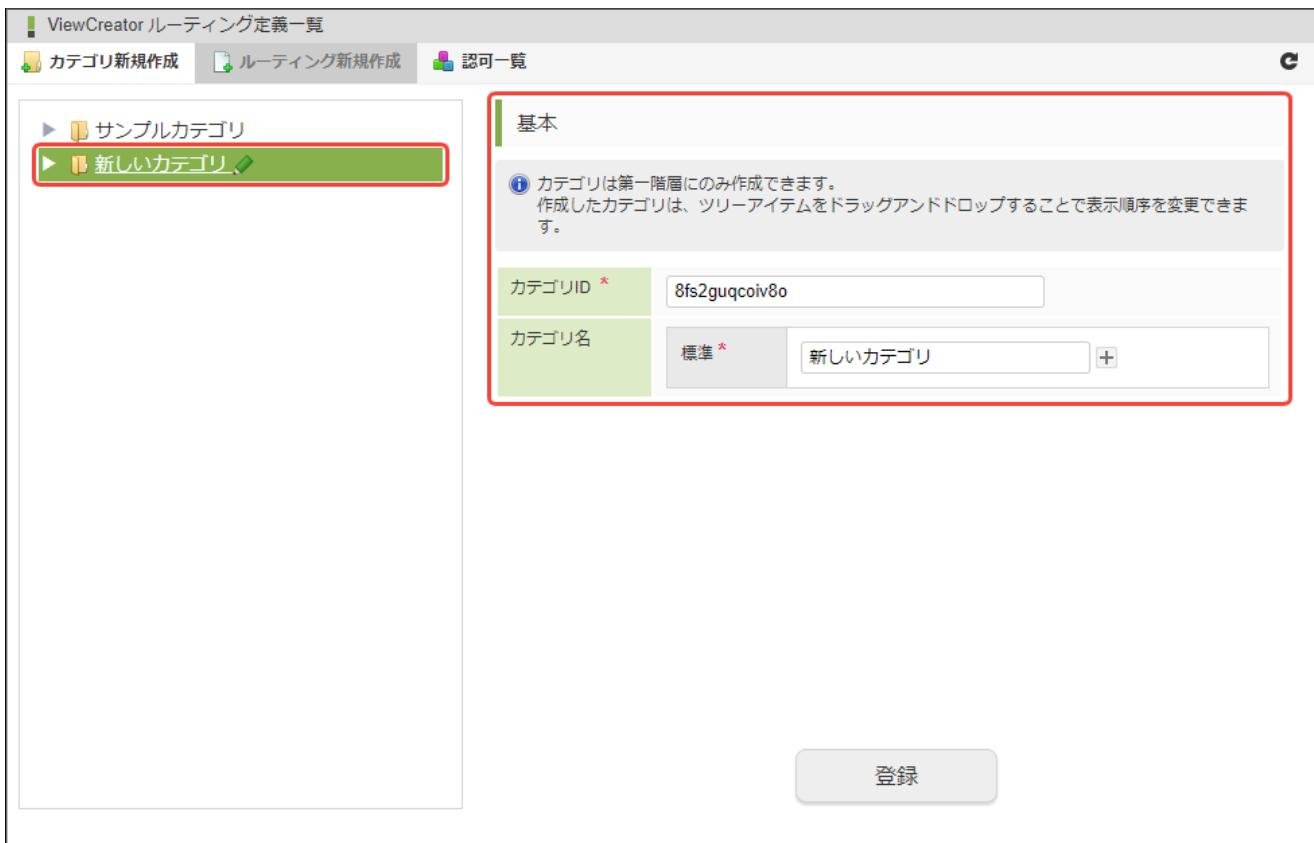
ルーティングはルーティングカテゴリに紐づけられる形で管理されているため、はじめにルーティングカテゴリを作成します。

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。

項目名	必須	説明
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。

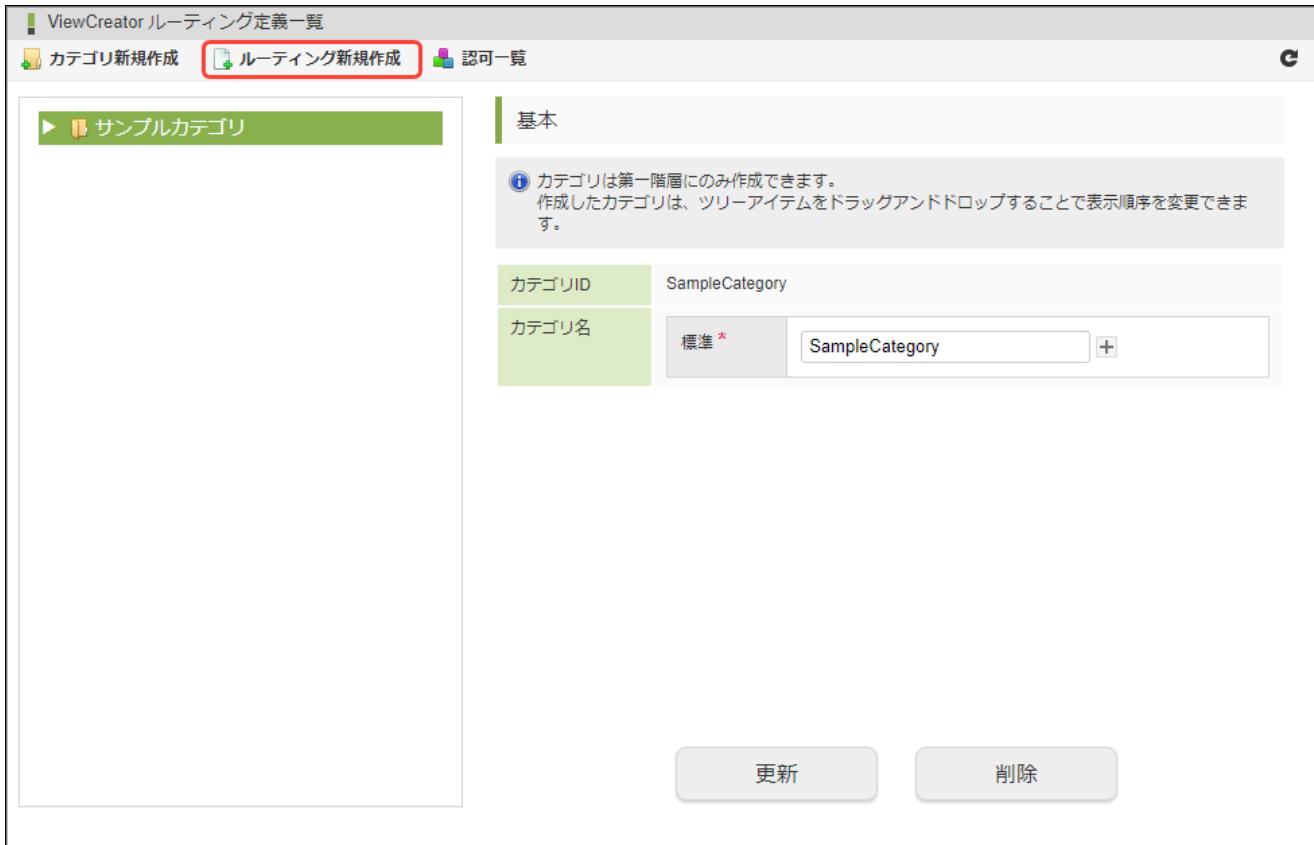
ルーティングの作成

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。

The screenshot shows the 'ViewCreator ルーティング定義一覧' (ViewCreator Routing Definition List) screen. On the left, there's a sidebar with 'サンプルカテゴリ' (Sample Category) expanded, showing 'サンプルルーティング' (Sample Routing) and '新しいルーティング' (New Routing). The '新しいルーティング' item is highlighted with a green box and has a red border around it. The main area is titled '基本' (Basic). It contains the following fields:

- カテゴリID**: SampleCategory
- ルーティングID ***: 8fscfpwo0uxv
- ルーティングURL ***: /imart/
- 対象クエリ ***: A dropdown menu with '検索' (Search), 'クエリコード' (Query Code), and 'クエリ名' (Query Name).
- メソッド ***: Radio buttons for 'GET' and 'POST'.
- 認証方式 ***: Radio buttons for 'IMAuthentication', 'OAuth', and 'Basic'.
- 認可URI**: viewcreator-query-rest://8fscfpwo0uxv
- レスポンス種別 ***: Radio buttons for 'JSONに変換して返却' (Convert to JSON and return) and 'CSVとして返却' (Return as CSV).
- ルーティング名**: A table with '標準 *' and '新しいルーティング'.
- 備考**: A table with '標準' and an empty text area.

A large red box surrounds the 'Basic' tab and the 'Routing ID' field area. At the bottom right is a '登録' (Register) button.

図：ルーティング情報

項目名	必須	説明
カテゴリID	—	カテゴリIDを表示します。 登録内容の変更はできません。
ルーティングID	必須	ルーティングを一意に表す文字列を設定します。 新規作成時のみ設定可能です。登録後の変更はできません。
ルーティングURL	必須	ルーティングを呼び出す際のURLを設定します。 他のルーティングと重複するURLは設定できません。 新規作成時のみ設定可能です。登録後の変更はできません。
対象クエリ	必須	実行対象のクエリを設定します。 「検索」をクリックして実行するクエリを選択してください。 詳細は「 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	—	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	—	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	—	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	—	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。

クエリ検索

クエリ名検索		
クエリコード	クエリ名	説明
5i7o4o3kli1pj	日本のデータ	
5i7ur3opijh2x	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	
5ib8ony9uhzy	transition_log	
8fh16zaswe6qchu	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

決定

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。

i 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。

The screenshot shows the 'Query Search' (クエリ検索) page. At the top, there is a search bar labeled 'クエリ名検索'. Below it is a table with columns: 'クエリコード' (Query Code), 'クエリ名' (Query Name), and '説明' (Description). The table contains six rows of data. At the bottom of the table, there is a pagination control with '1ページ中 1' (Page 1 of 1), 'ページ目' (Page Number), '50' (Number of items per page), and '5件中 1 - 5 を表示' (Showing 1 - 5 of 5). A large red rectangle highlights the '決定' (Decision) button at the bottom right of the table area.

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。



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。

The screenshot shows the 'ViewCreator ルーティング定義一覧' (Routing Definition List) screen. In the top navigation bar, there are tabs for 'カテゴリ新規作成' (Category New Creation), 'ルーティング新規作成' (Routing New Creation), and '認可一覧' (Authorization List). The 'ルーティング新規作成' tab is active.

The main area is titled 'ルーティング' (Routing) and contains several input fields:

- SampleCategory:** ルーティングID * (8h46brz1ukvix)
- URL:** ルーティングURL * (/imart/)
- クエリコード:** クエリコード (クエリ名)
- Method:** メソッド * (GET, POST)
- Authentication:** 認証方式 * (IMAuthentication, OAuth, Basic) - 'OAuth' is selected.
- OAuth Scope:** OAuth Scope * (検索 button is highlighted with a red box)
- Authorization URI:** 認可URI (viewcreator-query-rest://8h46brz1ukvix)
- Response Type:** レスポンス種別 * (JSONに変換して返却, CSVとして返却)
- Route Name:** ルーティング名 (標準, 新しいルーティング)
- Notes:** 備考 (標準)

At the bottom right, there is a '登録' (Register) button.

図：ルーティング情報 - OAuth スコープ

適用するスコープを選択

スコープ	表示名
core	コア機能へのアクセス
im-workflow-rest	IM-Workflow REST API へのアクセス
immaster	共通マスターへのアクセス
journal	履歴・コメントモジュール REST API へのアクセス
menu	メニューへのアクセス

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。

適用するスコープを選択

スコープ	表示名
core	コア機能へのアクセス
im-workflow-rest	IM-Workflow REST API へのアクセス
immaster	共通マスターへのアクセス
journal	履歴・コメントモジュール REST API へのアクセス
menu	メニューへのアクセス

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。

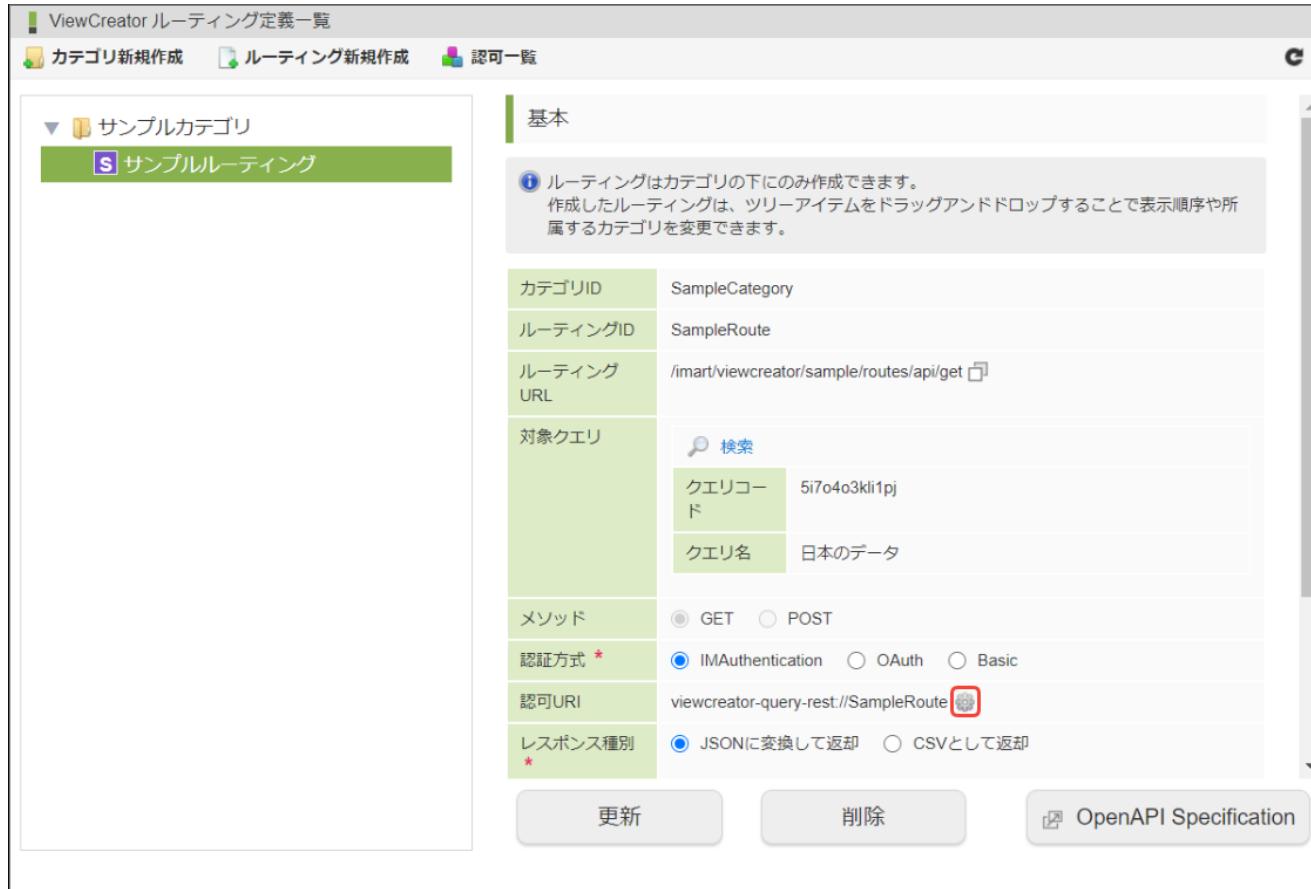
ルーティングの認可設定

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。

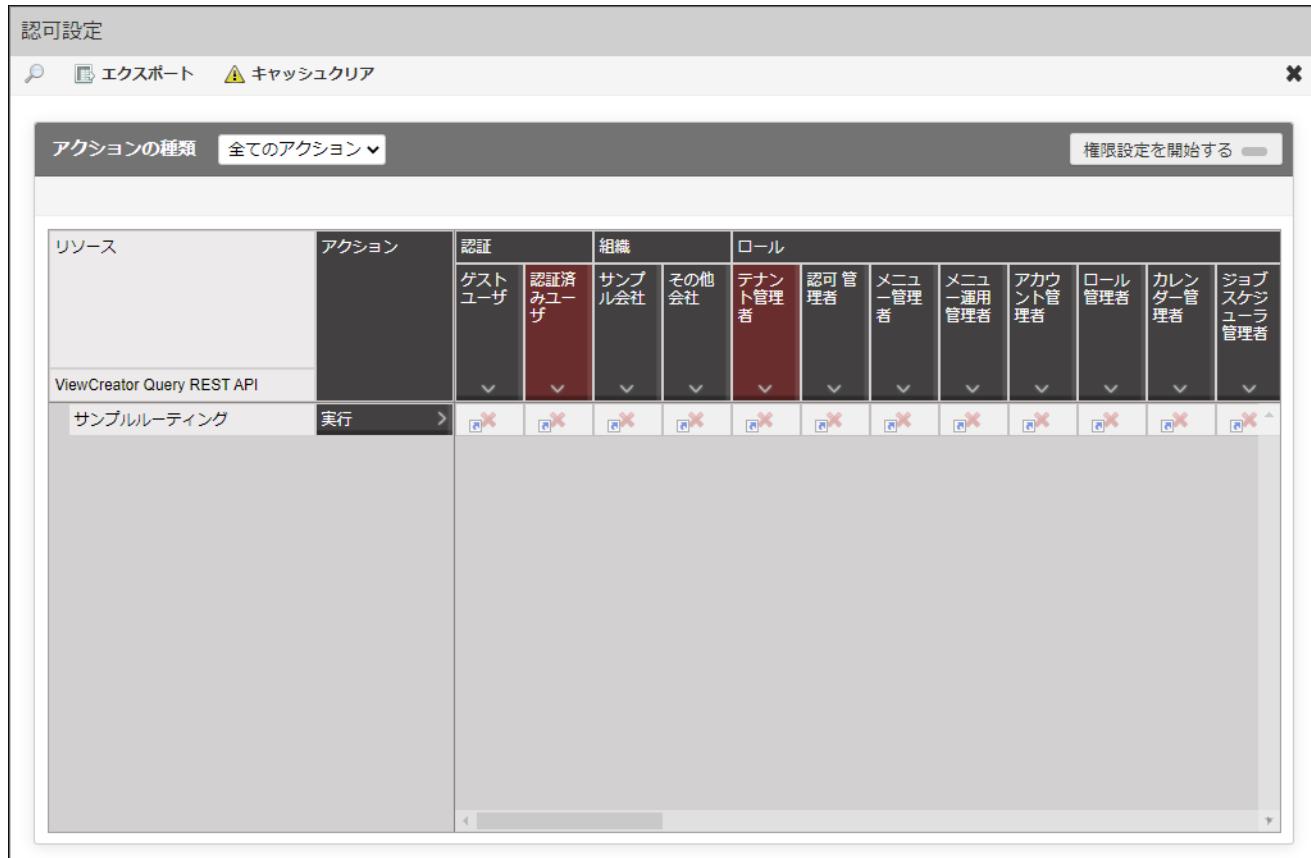
1. 認可設定をするルーティングをルーティングツリーから選択します。

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を利用して実行できます。

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。

ルーティング定義一覧							
カテゴリ新規作成	ルーティング新規作成						
認可一覧							
<div style="border: 1px solid #ccc; padding: 5px;"> ▼ サンプルカテゴリ </div> <div style="background-color: #0070C0; color: white; padding: 2px 10px; border-radius: 5px; display: inline-block;"> サンプルルーティング </div>							
ルーティングID	SampleRoute						
ルーティングURL	/imart/viewcreator/sample/routes/api/get						
対象クエリ	<div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;"> 🔍 検索 </div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">クエリコード</td> <td style="padding: 2px;">5i7o4o3kli1pj</td> </tr> <tr> <td style="padding: 2px;">クエリ名</td> <td style="padding: 2px;">日本のデータ</td> </tr> </table>	クエリコード	5i7o4o3kli1pj	クエリ名	日本のデータ		
クエリコード	5i7o4o3kli1pj						
クエリ名	日本のデータ						
メソッド	<input checked="" type="radio"/> GET <input type="radio"/> POST						
認証方式 *	<input checked="" type="radio"/> IMAuthentication <input type="radio"/> OAuth <input type="radio"/> Basic						
認可URI	viewcreator-query-rest://SampleRoute						
レスポンス種別 *	<input checked="" type="radio"/> JSONに変換して返却 <input type="radio"/> CSVとして返却						
ルーティング名	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">標準 *</td> <td style="padding: 2px;"><input type="text" value="SampleRouting"/></td> </tr> <tr> <td style="padding: 2px;">英語</td> <td style="padding: 2px;"><input type="text" value="SampleRouting"/></td> </tr> <tr> <td style="padding: 2px;">日本語</td> <td style="padding: 2px;"><input type="text" value="サンプルルーティング"/></td> </tr> </table>	標準 *	<input type="text" value="SampleRouting"/>	英語	<input type="text" value="SampleRouting"/>	日本語	<input type="text" value="サンプルルーティング"/>
標準 *	<input type="text" value="SampleRouting"/>						
英語	<input type="text" value="SampleRouting"/>						
日本語	<input type="text" value="サンプルルーティング"/>						
更新 削除 OpenAPI Specification							

図：「ルーティング定義一覧」画面 - OpenAPIを開く

ViewCreator クエリ実行 API

8.0.27 OAS3

ViewCreator で定義されたクエリを実行するAPI仕様です。

Server

<http://localhost:8080/imart>

サンプルルーティング

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

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

図：OpenAPI REST APIビューア

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

実行するクエリのカラム定義や抽出条件値定義を取得します。

Parameters

No parameters

Execute **Cancel** **Clear**

Responses

Curl

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

Request URL

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

Server response

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

図：REST API実行 - クエリのメタ情報取得

The screenshot shows the ViewCreator API Explorer interface. At the top, a blue header bar indicates a **GET /viewcreator/sample/routes/api/get** query to execute. Below this, a message says "クエリを実行して結果を取得します。" (Execute the query to get the results). A "Parameters" section shows "No parameters". Below that are "Execute" and "Clear" buttons. The "Responses" section includes a "Curl" command:

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

The "Request URL" is listed as <http://localhost:8080/imart/viewcreator/sample/routes/api/get>. The "Server response" section shows a 200 status code. The "Details" tab displays the JSON response body:

```
{
  "data": [
    {
      "year": "1970",
      "name": "北海道",
      "cccd00003": "北海道",
      "age": "0 - 14歳",
      "area": "83455.33",
      "population": "1309"
    }
  ]
}
```

図：REST API実行 - クエリの実行結果取得

REST API 仕様

クエリを実行するREST APIの動作仕様について説明します。

セッション管理

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

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

ヘッダを省略した場合は *once* を指定した場合と同じです。

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

- セッション管理

リクエストパラメータ

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

パラメータ	データ型	説明
<i>limit</i>	数値型	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: <i>limit=10</i>

パラメータ	データ型	説明
offset	数値型	<p>レコードの取得開始位置を設定します。</p> <p>省略した場合は、デフォルト設定（0）が適用されます。</p> <p>設定例: <code>offset=10</code></p>
columns	文字列型	<p>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。</p> <p>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。</p> <p>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。</p> <p>設定例: <code>columns=user_cd,ccd00003,user_name</code></p>
sort	文字列型	<p>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。</p> <p>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。</p> <p><code>order</code> パラメータと組み合わせて使用します。</p> <p>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。</p> <p>設定例: <code>sort=user_cd&order=asc</code></p>
<div style="border: 1px solid #ccc; padding: 10px;"> <p>i コラム</p> <p><code>order</code> パラメータを設定しないことで、クエリで定義されているソート条件を除去できます。</p> </div>		
order	文字列型	<p>第1ソートキー（<code>sort</code>）に対するソート方向を設定します。</p> <p><code>asc</code> または <code>desc</code> を設定します。</p> <p>設定例: <code>sort=user_cd&order=desc</code></p>
secondSort	文字列型	<p>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。</p> <p>詳細は <code>sort</code> パラメータの説明を参照してください。</p>
secondOrder	文字列型	<p>第2ソートキー（<code>secondSort</code>）に対するソート方向を設定します。</p> <p>詳細は <code>order</code> パラメータの説明を参照してください。</p>
thirdSort	文字列型	<p>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。</p> <p>詳細は <code>sort</code> パラメータの説明を参照してください。</p>
thirdOrder	文字列型	<p>第3ソートキー（<code>thirdSort</code>）に対するソート方向を設定します。</p> <p>詳細は <code>order</code> パラメータの説明を参照してください。</p>
responseKeyType	文字列型	<p>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。</p> <p>以下のいずれかを設定できます。</p> <ul style="list-style-type: none"> ▪ <code>NAME</code> 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>CODE</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。 ▪ <code>CAPTION</code> 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。 <p>省略した場合は、デフォルト設定（<code>NAME</code>）が適用されます。</p> <p>設定例: <code>responseKeyType=NAME</code></p>
<div style="background-color: #fffacd; border: 1px solid #ffcc99; padding: 10px;"> <p>! 注意</p> <p><code>CODE</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。</p> </div>		
metadata	—	<p>クエリ定義のメタ情報を取得する場合に設定します。</p> <p>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。</p> <p>詳細は「レスポンスボディ」を参照してください。</p>

i 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。

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。 <i>limit</i> パラメータや <i>offset</i> 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の標準インター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。

The screenshot shows the 'ViewCreator ルーティング定義一覧' (ViewCreator Routing Definition List) screen. On the left, there's a tree view under 'サンプルカテゴリ' (Sample Category) with 'サンプルルーティング' (Sample Routing) selected, which is highlighted with a red box. The main panel displays routing details for 'SampleRoute'. It includes fields for 'ルーティングID' (Routing ID), 'ルーティングURL' (Routing URL), '対象クエリ' (Target Query), 'メソッド' (Method), '認証方式' (Authentication Method), '認可URI' (Authorization URI), 'レスポンス種別' (Response Type), and 'ルーティング名' (Routing Name). At the bottom right, there are '更新' (Update), '削除' (Delete), and 'OpenAPI Specification' buttons, with the 'OpenAPI Specification' button also highlighted with a red box.

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。

The screenshot shows the 'ViewCreator クエリ実行 API' (ViewCreator Query Execution API) viewer. At the top, it says 'ViewCreator で定義されたクエリを実行するAPI仕様です。' (This is the API specification for executing queries defined in ViewCreator). Below this, there's a 'Server' dropdown set to 'http://localhost:8080/imart'. A section titled 'サンプルルーティング' (Sample Routing) contains two API endpoints: 'GET /viewcreator/sample/routes/api/get?metadata メタデータ取得API' and 'GET /viewcreator/sample/routes/api/get クエリ実行API'. Both of these endpoints are highlighted with red boxes.

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。

ViewCreator クエリ実行 API 8.0.27 OAS3

ViewCreator で定義されたクエリを実行するAPI仕様です。

Server

サンプルルーティング ▼

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

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

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。

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

クエリを実行して結果を取得します。

Parameters

Try it out

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

図：REST API実行 - 「Try it out」のクリック

画面項目	説明
Parameters	REST APIの要求するパラメータの詳細です。 ここでは以下が確認できます。詳細は「 リクエストパラメータ 」を参照してください。 <ul style="list-style-type: none"> ■ 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

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

Execute

図：REST API実行 - 「Execute」のクリック

6. 実行結果が表示されます。

Curl

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

Request URL

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

Server response

Code	Details
200	<p>Response body</p> <pre>{ "data": [{ "age": "0 ~ 14歳", "year": "1970", "area": "1862-01" }, { "age": "0 ~ 14歳", "year": "1970", "area": "1893-76" }], "totalCount": 987 }</pre> <p>Download</p> <p>Response headers</p> <pre>content-length: 130 content-type: application/json date: Tue 02 Mar 2021 01:00:44 GMT server: Resin/4.0.64 x-jp-co-intra-mart-viewcreator-total-count: 987</pre>

図：REST API実行 - 実行結果

画面項目	説明
Curl	Curlを用いて対象のREST APIを実行する場合のコマンドです。
Request URL	実行時にリクエストを送信したURLです。
Code	REST APIを実行した結果として返却されたHTTPステータスコードです。
Response body	レスポンスの結果（Body情報）です。各出力データの形式の詳細は「 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。

ViewCreator ルーティングサンプル

年度 開始 終了

地域名

検索

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

1 2 3 4 5 ... 99
Previous Next page

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。

The screenshot shows the 'クエリ一覧' (Query List) screen. At the top, there are buttons for '新規' (New), 'コピー' (Copy) (which is highlighted with a red box), '削除' (Delete), and 'データ参照一覧へ' (List of Data References). Below the header is a search bar with '検索' (Search) and 'クリア' (Clear) buttons. The main area is a table listing queries:

□ データ参照作成	クエリ名	クエリコード	接続 ID	更新日	作成者	エクスポート
<input type="checkbox"/>	system_log	5ib8yonru57ws	files	6月 12, 2014 10:58 午前	system	
<input type="checkbox"/>	transition_log	5ib8yony9uhzy	files	6月 12, 2014 10:58 午前	system	
<input type="checkbox"/>	ViewCreatorの設定テーブルを利用	5i7ur3opljh2x	default	11月 6, 2013 2:40 午前	tenant	
<input checked="" type="checkbox"/>	日本のデータ	5i7o4o3kli1pj	default	10月 29, 2013 2:26 午後	tenant	
<input type="checkbox"/>	SQL 日本のデータ (データ結合)	8fh16zaswe6qchu	default	2月 6, 2020 3:28 午後	tenant	

At the bottom of the table, there are navigation buttons: 1 ページ中 1 ページ目 50 件中 1 - 5 を表示.

Below the table are buttons for 'ファイル追加...' (File Add), '中断' (Interrupt), '削除' (Delete), and 'インポート' (Import).

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。

The screenshot shows the 'クエリ一覧' (Query List) screen, similar to the previous one. The query 'コピー～日本のデータ' is now highlighted with a red box. The rest of the interface is identical to the first screenshot.

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

上へ	下へ	テーブル/ビュー/リソース	カラム/フィールド	型	ソート	キャプション	削除
		sample_population	year	数値	指定なし	year	
		sample_region	name	文字列	指定なし	regionName	
		sample_prefectu	name	文字列	指定なし	prefectureName	
		sample_age	age	文字列	指定なし	age	
		sample_prefectu	area	数値	指定なし	area	
		sample_population	population	数値	指定なし	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。

上へ	下へ	テーブル/ビュー/リソース	カラム/フィールド	型	ソート	キャプション	削除
		sample_population	year	数値	指定なし	year	
		sample_region	name	文字列	指定なし	regionName	
		sample_prefectu	name	文字列	指定なし	prefectureName	
		sample_age	age	文字列	指定なし	age	
		sample_prefectu	area	数値	指定なし	area	
		sample_population	population	数値	指定なし	population	

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）。

上へ	下へ	テーブル/ビュー/リソース	カラム/フィールド	抽出方法	条件値	削除
		sample_population	year	完全一致		

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)%>`

抽出条件一覧					
上へ	下へ	テーブル/ビュー/リソース	カラム/フィールド	抽出方法	条件値
		sample_population	year	以上	<%REQUEST_PARAMETER(minimumYearParameter)%>

図：「クエリ編集」画面 - 抽出条件設定

8. 同様に、以下のとおりに抽出条件を追加します。

- テーブル/ビュー/リソース : `sample_population`
- カラム/フィールド : `year`
- 抽出方法 : 「以下」
- 条件値 : `<%REQUEST_PARAMETER(maxYearParameter)%>`

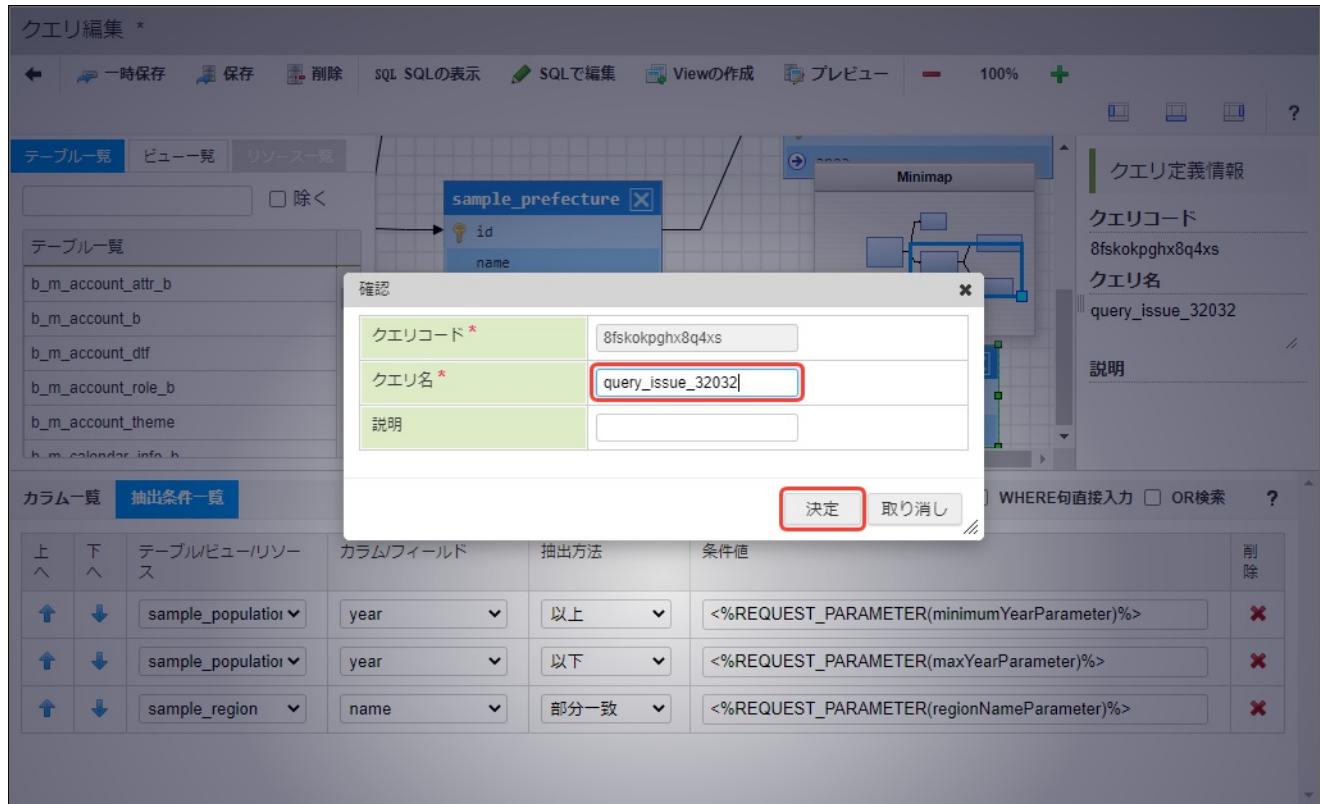
- テーブル/ビュー/リソース : `sample_region`
- カラム/フィールド : `name`
- 抽出方法 : 「部分一致」
- 条件値 : `<%REQUEST_PARAMETER(regionNameParameter)%>`

抽出条件一覧					
上へ	下へ	テーブル/ビュー/リソース	カラム/フィールド	抽出方法	条件値
		sample_population	year	以上	<%REQUEST_PARAMETER(minimumYearParameter)%>
		sample_population	year	以下	<%REQUEST_PARAMETER(maxYearParameter)%>
		sample_region	name	部分一致	<%REQUEST_PARAMETER(regionNameParameter)%>

図：「クエリ編集」画面 - 抽出条件追加設定

9. 「クエリ編集」画面上部の「保存」ボタンをクリックします。

10. 「確認」ダイアログでクエリ名を「query_issue_32032」に変更して「決定」をクリックします。



図：「クエリ編集」画面 - クエリ保存

ルーティング定義の作成

次に、ルーティング定義を作成します。

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



図：「ルーティング定義一覧」画面 - 「ルーティング新規作成」

3. ルーティング情報の各項目を以下のとおりに設定します。

- ルーティングID : *routing_issue_32032*
- ルーティングURL : *viewcreator/sample/routes/api/get*
- 対象クエリ : 「[実行するクエリの作成](#)」で作成したクエリ定義
- メソッド : 「GET」
- 認証方式 : 「IMAAuthentication」
- レスポンス種別 : 「JSONに変換して返却」
- ルーティング名 : *SampleRouting(issue32032)*

ViewCreator ルーティング定義一覧

カテゴリ新規作成 ルーティング新規作成 認可一覧

▼ ViewCreator サンプル
R サンプルルーティング (issue32032)

ルーティングID *	routing_issue_32032
ルーティングURL *	/imart/viewcreator/sample/routes/api/get
対象クエリ *	<input type="button" value="検索"/> 検索 クエリコード: 8fwbgn85rpwt65f クエリ名: query_issue_32032
メソッド *	<input checked="" type="radio"/> GET <input type="radio"/> POST
認証方式 *	<input checked="" type="radio"/> IMAuthentication <input type="radio"/> OAuth <input type="radio"/> Basic
認可URI	viewcreator-query-rest://routing_issue_32032
レスポンス種別 *	<input checked="" type="radio"/> JSONに変換して返却 <input type="radio"/> CSVとして返却
ルーティング名	標準 *: SampleRouting(issue32032) 英語: SampleRouting(issue32032) 日本語: サンプルルーティング (issue32032)

登録

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。

ViewCreator ルーティング定義一覧

カテゴリ新規作成 ルーティング新規作成 認可一覧

▼ ViewCreator サンプル
R サンプルルーティング (issue32032)

ルーティングID	routing_issue_32032
ルーティングURL	/imart/viewcreator/sample/routes/api/get
対象クエリ	<input type="button" value="検索"/> 検索 クエリコード: 8fwbgn85rpwt65f クエリ名: query_issue_32032
メソッド	<input checked="" type="radio"/> GET <input type="radio"/> POST
認証方式 *	<input checked="" type="radio"/> IMAuthentication <input type="radio"/> OAuth <input type="radio"/> Basic
認可URI	viewcreator-query-rest://routing_issue_32032
レスポンス種別 *	<input checked="" type="radio"/> JSONに変換して返却 <input type="radio"/> CSVとして返却
ルーティング名	標準 *: SampleRouting(issue32032) 英語: SampleRouting(issue32032) 日本語: サンプルルーティング (issue32032)

更新 削除 OpenAPI Specification

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。

リソース	アクション	認可		組織		ロール																
		ゲストユーザ	認可済みユーザ	サンプル会社	その他会社	テナント管理者	認可管理者	メニュー管理者	メニュー運用管理者	アカウント管理者	ロール管理者	カレンダーマネージャー	ジョブスケジューラマネージャー									
ViewCreator Query REST API	実行	<	>	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
サンプルルーティング	実行	<	>	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×

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。

認可にズレがある場合、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>

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。

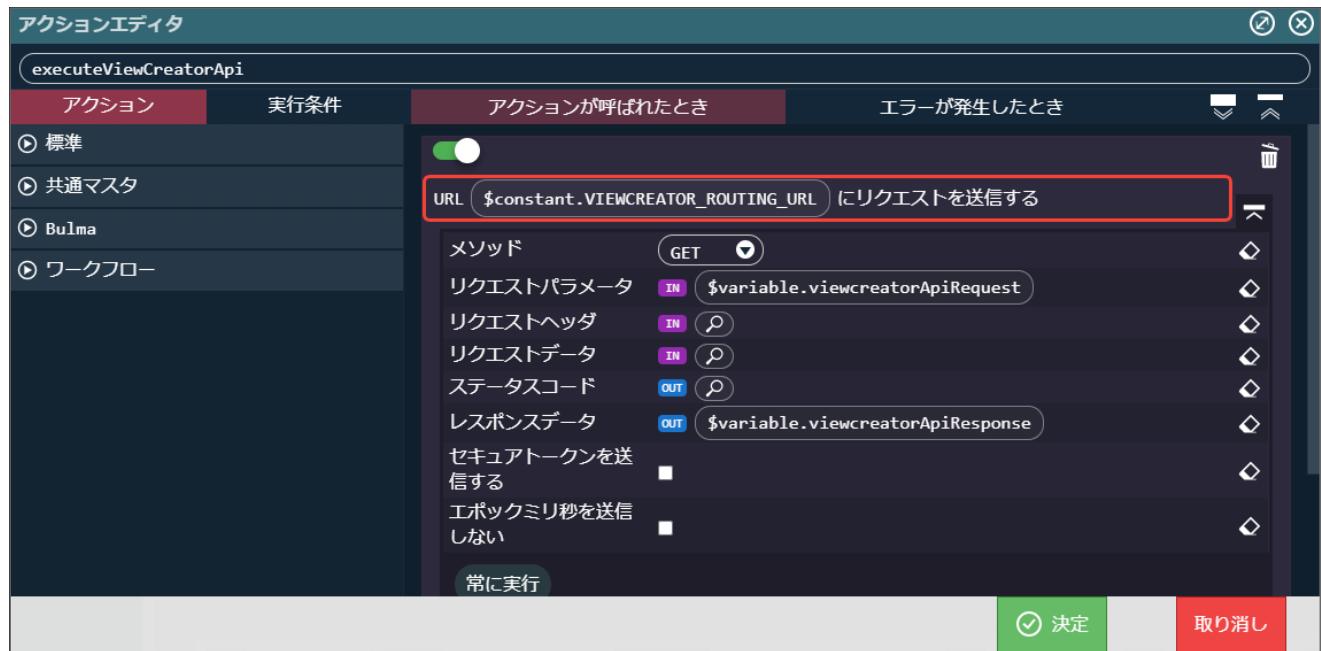
■ REST APIの実行

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"
}
```

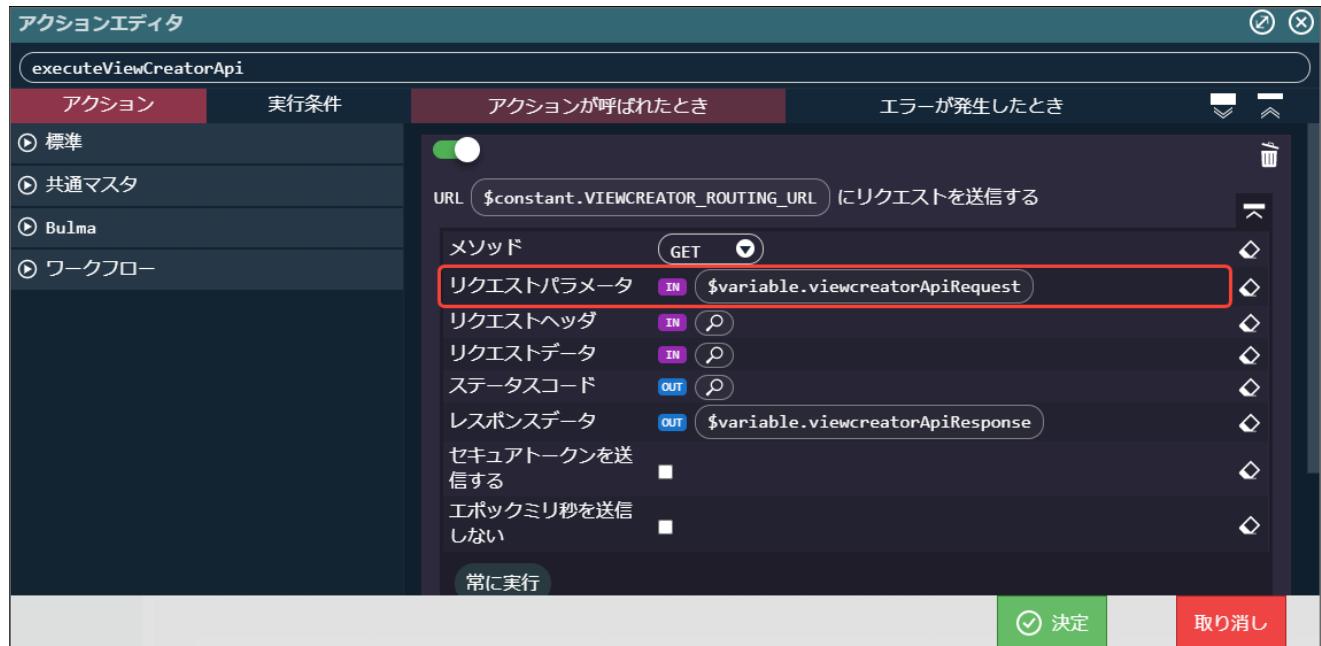


図：executeViewCreatorApi アクション

■ 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。



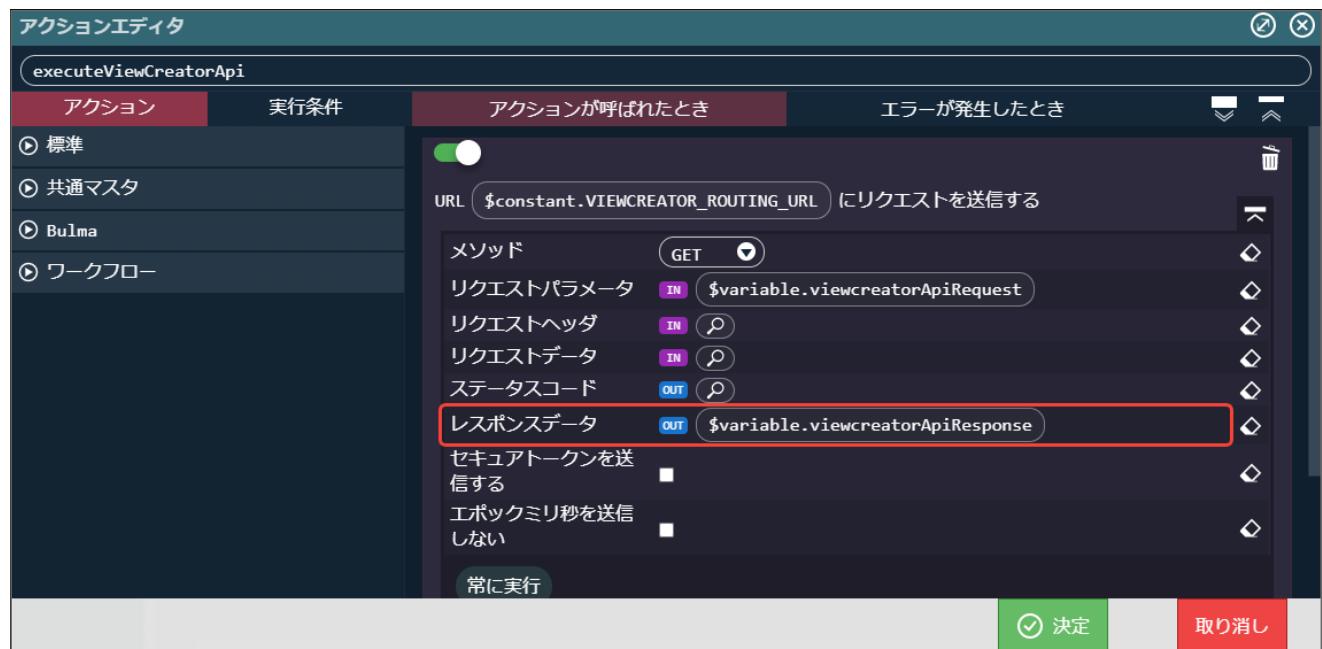
図：executeViewCreatorApi アクション - リクエストパラメータ

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

■ REST APIの実行結果と画面の連携

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

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

ViewCreator SQLビルダ アシスタントによるSQL生成

ViewCreator SQLビルダ アシスタントは生成AIを利用してユーザのSQL作成をサポートします。



コラム

ViewCreator SQLビルダ アシスタントは 2024 Autumn(Jasmine) から利用可能です。

前提条件

ViewCreator SQLビルダ アシスタント（以下 アシスタント）を利用するには、ViewCreator Copilot をモジュールに追加してセットアップをしてください。

セットアップについては「[IM-Copilot 利用ガイド](#)」 - 「[ViewCreator SQLビルダ アシスタントのセットアップ](#)」を参照してください。



コラム

ViewCreator Copilot は IM-Copilot がご利用可能なエディションに準拠してご利用できます。

IM-Copilot がご利用可能なエディションについては「[IM-Copilot 利用ガイド](#)」 - 「[セットアップ（iAP）](#)」を参照してください。



注意

本機能は、AIによりチャット形式で応答やSQLを生成するものです。

生成される内容については正確性・完全性・目的適合性を保証するものではなく、意図しない応答や目的外の情報が含まれる場合があります。

生成物の著作権はユーザに帰属します。

利用者は、生成された応答や成果物を参考情報として取り扱い、社内利用・外部公開のいずれにおいても、内容の妥当性や安全性を確認したうえで、自己の責任で利用してください。

外部公開や配布を行う場合は、社内で適切なレビューを行い、第三者著作物やライセンス条件を遵守してください。

準備

アシスタントを利用するには、認可の設定とジョブネットの実行が必要です。

認可の設定

アシスタントを利用するには、認可設定において「SQLビルダ」と「ViewCreator SQLビルダ アシスタント」の両方が許可されている必要があります。

設定方法の詳細は「[認可の設定](#)」、「[ViewCreator SQLビルダ アシスタントを利用する場合](#)」を参照してください。

ジョブネットの実行

アシスタントはテーブルメタデータ情報を使用してSQLを生成します。

初めてアシスタントを利用する場合は、次のジョブネットを実行してテーブルメタデータ情報をベクトルデータベースに登録してください。



ジョブネットカテゴリ	ジョブネット名
ViewCreator Copilot	テーブルメタデータ同期+ベクトルデータベース登録

ジョブネットの詳細は「[ジョブ・ジョブネットリファレンス](#)」 - 「[テーブルメタデータ同期+ベクトルデータベース登録](#)」を参照してください。

独自テーブルのテーブルメタデータ情報を取り込む場合

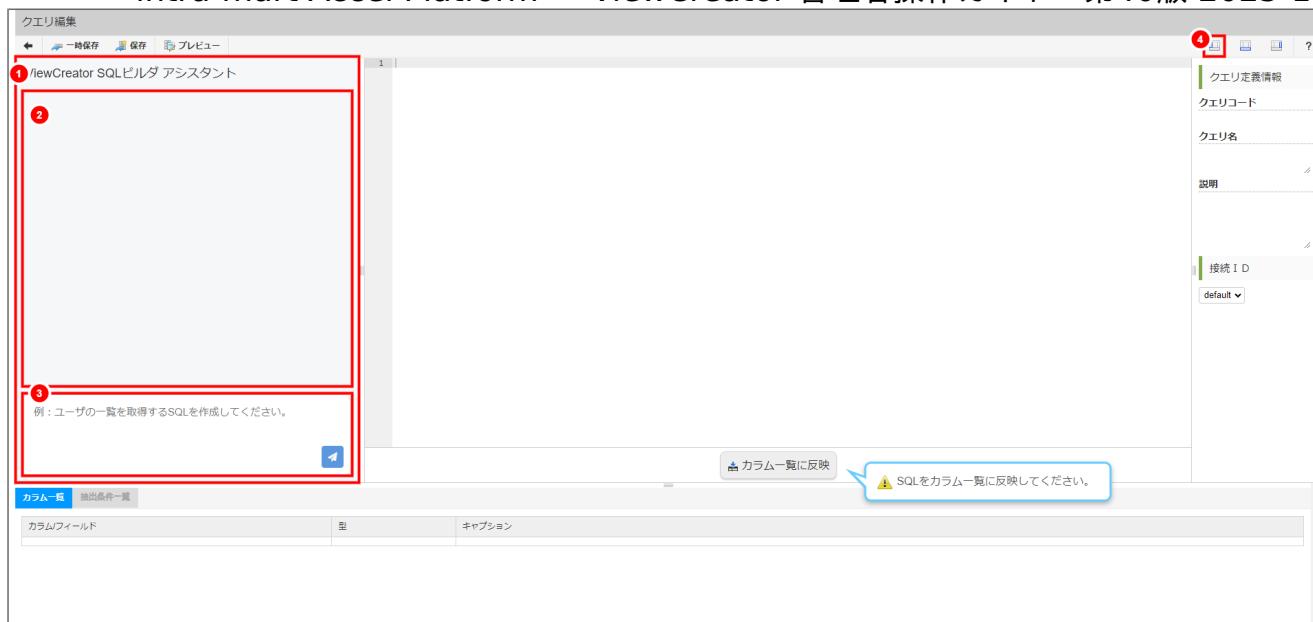
独自テーブルのテーブルメタデータ情報を取り込む場合は「[独自のテーブルメタデータ情報を追加する](#)」を参照してください。

ViewCreator SQLビルダ アシスタント

アシスタントはクエリ編集画面内に表示されます。

クエリ編集画面の操作方法については、「[SQLビルダによるクエリの作成](#)」を参照してください。

アシスタントの具体的な利用方法については、「[ViewCreator SQLビルダ アシスタントの利用例](#)」を参照してください。



図：ViewCreator SQLビルダ アシスタント - 初期表示

1. アシスタント 領域です。
2. ユーザが送信した質問と アシスタント からの回答が表示される領域です。
3. ユーザが質問を入力するテキストエリアです。送信ボタンをクリックすると入力内容が送信されます。
4. アシスタント 領域の表示/非表示を切り替えます。

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。

アクセス種別-認可の利用について



コラム

アクセス種別-認可は「[アクセス種別-認可を利用する場合](#)」が設定されたユーザが利用可能です。

「作成者」または「管理者」リソースが許可されているユーザの場合

アクセス種別-参照権のデータ参照の場合、参照権が未設定の状態は「作成者」または「管理者」リソースが許可されているユーザがデータ参照画面を表示できます。

アクセス種別-認可のデータ参照の場合、認可が未設定の状態（データ参照登録直後）は認可リソースグループ「データ参照のアクセス権設定」が許可されているユーザ（標準では「テナント管理者」「ViewCreator管理者」ロール）がデータ参照画面を表示できます。

「作成者」または「管理者」リソースが許可されているユーザでもアクセス種別-認可のデータ参照は認可設定がなければデータ参照画面が表示できません。

参照できないデータ参照は編集や削除、インポートによる更新もできません。

アクセス種別-認可のデータ参照の参照判定時には認可機能に問い合わせる必要があるため、アクセス種別-認可を設定しているデータ参照が増加すると、「データ参照一覧」画面等の全データ参照の参照判定を必要とする処理では表示にかかる時間が増加します。同様に「インポート」画面で大量のアクセス種別-認可のデータ参照をインポートすると、インポートの処理速度は低下します。参照権の設定で実現できる場合はアクセス種別-参照権を設定する、不要なデータ参照は削除するなどの運用を行うことで、「データ参照一覧」画面表示等の低速化を予防できます。

アクセス種別-認可を設定できないユーザがコピー、インポートを行う場合

アクセス種別-認可を設定できないユーザがコピーやインポートを行う際は以下の通り制限されます。

- アクセス種別-認可を設定できないユーザがアクセス種別-認可のデータ参照をコピーした場合、アクセス種別-参照権としてコピーされます。
- アクセス種別-認可を設定できないユーザがアクセス種別-認可のインポート資材を新規インポートした場合、アクセス種別-参照権としてインポートされます。

アクセス種別-認可を設定できないユーザのデータ参照をアクセス種別-認可に変更した場合

アクセス種別-認可を設定できないユーザが作成したデータ参照をアクセス種別-認可に変更した場合、以下の様に作成者にもアクセス種別項目が表示されます。



権限設定

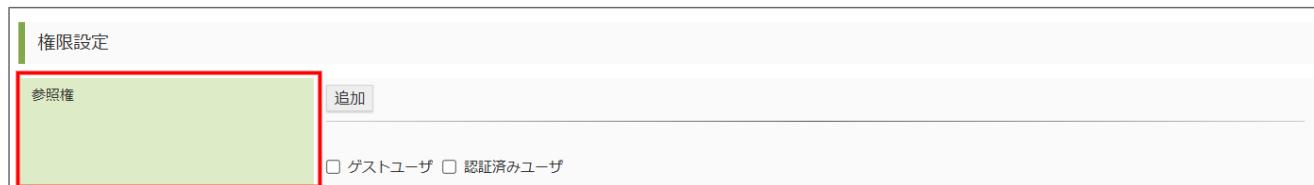
アクセス種別

○ 参照権 認可

認可

vc-dataview://5i7urzmesvqqip4

その後アクセス種別-参照権に変更すると、作成者にアクセス種別項目は表示されなくなります。



権限設定

参照権

追加

ゲストユーザ 認証済みユーザ

2025 Spring(Kamille)より前のバージョンのソースコード出力を利用している場合

2025 Spring(Kamille)より前のバージョンのソースコード出力で作成したデータ参照画面を利用して、2025 Spring(Kamille)以上のバージョンに上げた際、画面表示時に以下のWARNログが出力される場合があります。

```
[MSG.W.IWP.VIEWCREATOR.DATAVIEWEDIT.ACCESS.TYPE.EMPTY] アクセス種別が空です。
```

ログを抑制したい場合は対象のデータ参照定義の <root> 配下に以下を追加してください。

```
<accessType datatype="string">READABLES</accessType>
```

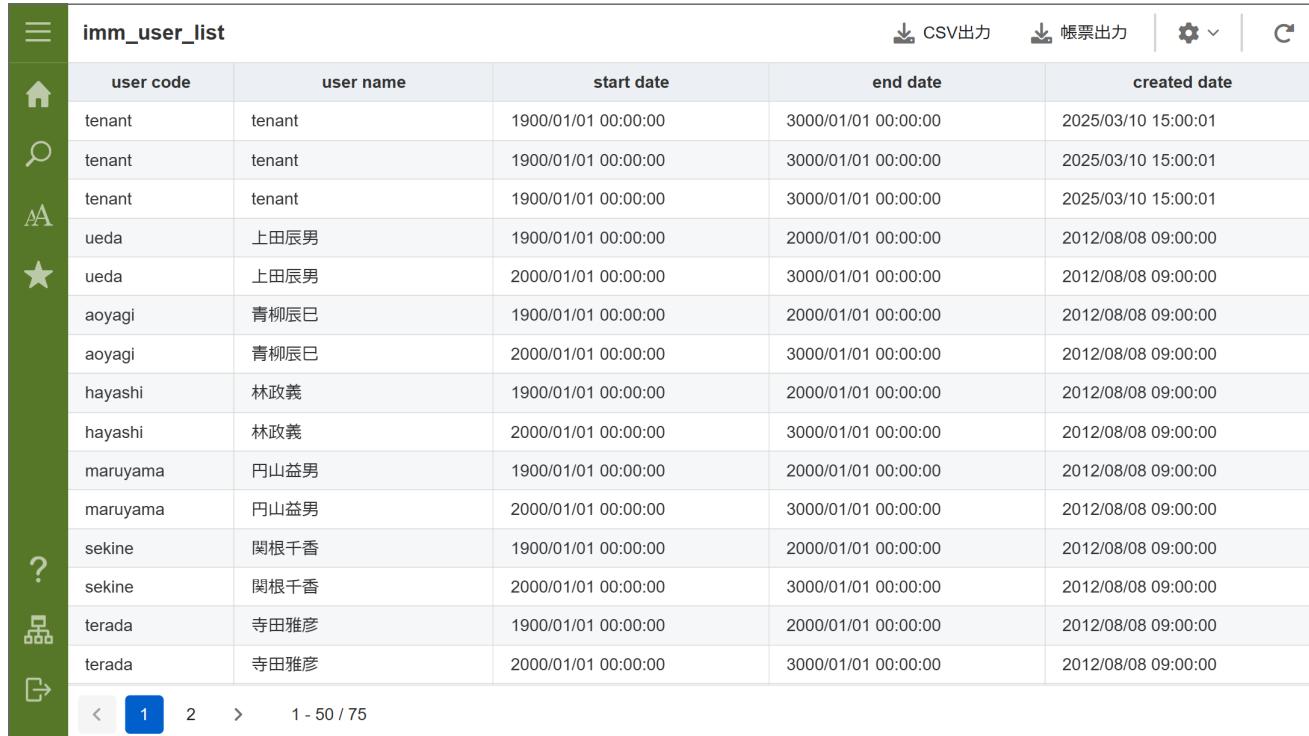
付録

チュートリアル

IM-LogicDesigner を利用して取得したタイムスタンプ型のレコードを ViewCreator のデータ参照のリスト集計で表示する。

このチュートリアルでは、IM-LogicDesigner を利用して取得したタイムスタンプ型のレコードを ViewCreator のデータ参照のリスト集計で表示する例を紹介します。

具体的には、IM-LogicDesigner のロジックフローを利用してデータベースから取得したユーザデータを一覧として表示する画面を作成します。



imm_user_list					CSV出力	帳票出力	▼	
	user code	user name	start date	end date	created date			
tenant	tenant		1900/01/01 00:00:00	3000/01/01 00:00:00	2025/03/10 15:00:01			
tenant	tenant		1900/01/01 00:00:00	3000/01/01 00:00:00	2025/03/10 15:00:01			
tenant	tenant		1900/01/01 00:00:00	3000/01/01 00:00:00	2025/03/10 15:00:01			
ueda	上田辰男		1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00			
ueda	上田辰男		2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00			
aoyagi	青柳辰巳		1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00			
aoyagi	青柳辰巳		2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00			
hayashi	林政義		1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00			
hayashi	林政義		2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00			
maruyama	円山益男		1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00			
maruyama	円山益男		2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00			
sekine	関根千香		1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00			
sekine	関根千香		2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00			
terada	寺田雅彦		1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00			
terada	寺田雅彦		2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00			

図：完成イメージ



注意

タイムスタンプ型および日付型のレコードをユーザごとのアカウントタイムゾーンを考慮して表示したい場合は、「[intra-mart Accel Platform タイムゾーン仕様書](#)」を参照した上で本チュートリアルをお試しください。
また、本サンプルのアカウントタイムゾーンは「America/Los_Angeles」です。



コラム

IM-LogicDesigner の詳細については、「[IM-LogicDesigner仕様書](#)」を参照してください。

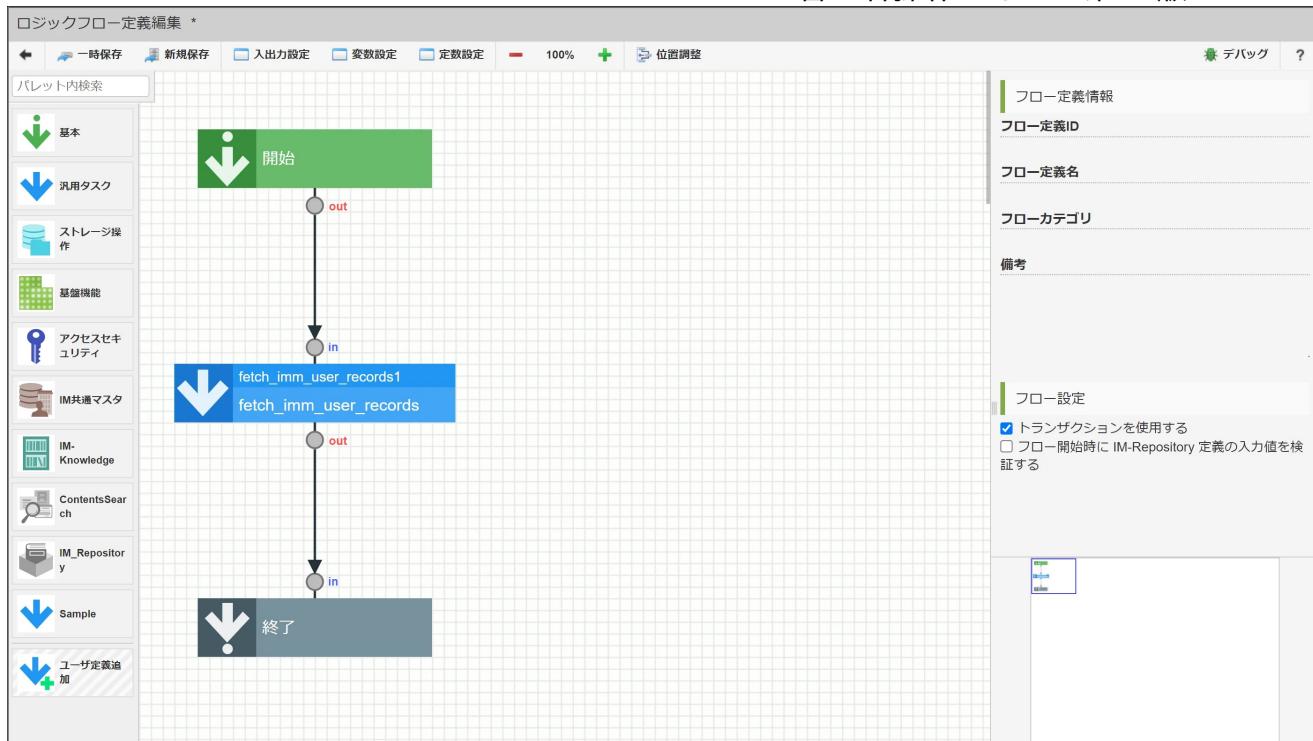
目次

- ロジックフローを作成する
- ロジックフローを ViewCreator で利用する
- データ参照のリスト集計を作成する
- 作成したデータ参照を閲覧する

ロジックフローを作成する

はじめに、ViewCreator で利用するロジックフローを作成します。

具体的には、SQLユーザ定義を利用してデータベースのレコードを取得するロジックフローを作成します。



図：ロジックフロー

1. まず、データベースからレコードを取得するためのSQL定義を作成します。
ユーザ定義の新規作成手順は「[ユーザ定義を新規登録する](#)」を参照してください。
2. 「SQL定義編集」画面を表示したら、「ユーザ定義共通設定」を以下のように設定します。

項目名	設定値
ユーザ定義ID	fetch_imm_user_records
ユーザ定義名：標準	fetch_imm_user_records
ユーザカテゴリ	任意のものを選択してください。

3. 次に、「SQL定義」を以下のように設定します。

SQL 定義

データベース種別 *: TENANT

クエリ種別 *: SELECT

クエリ *

```
① データ定義を取得する
1 SELECT
2   user_cd,
3   start_date,
4   end_date,
5   user_name,
6   create_date
7   FROM
8     imm_user
```

図：「SQL定義編集」画面 - SQL定義の設定

項目名	設定値
データベース種別	問い合わせしたいデータベースを選択してください。 今回は、TENANT を選択します。
クエリ種別	SELECT

次に、「クエリ」を以下のように設定します。

```
SELECT
user_cd,
start_date,
end_date,
user_name,
create_date
FROM
imm_user
```

4. 「クエリ」が設定できたら、「データ定義を取得する」をクリックし、「クエリ」をもとに生成される「返却値」を反映します。

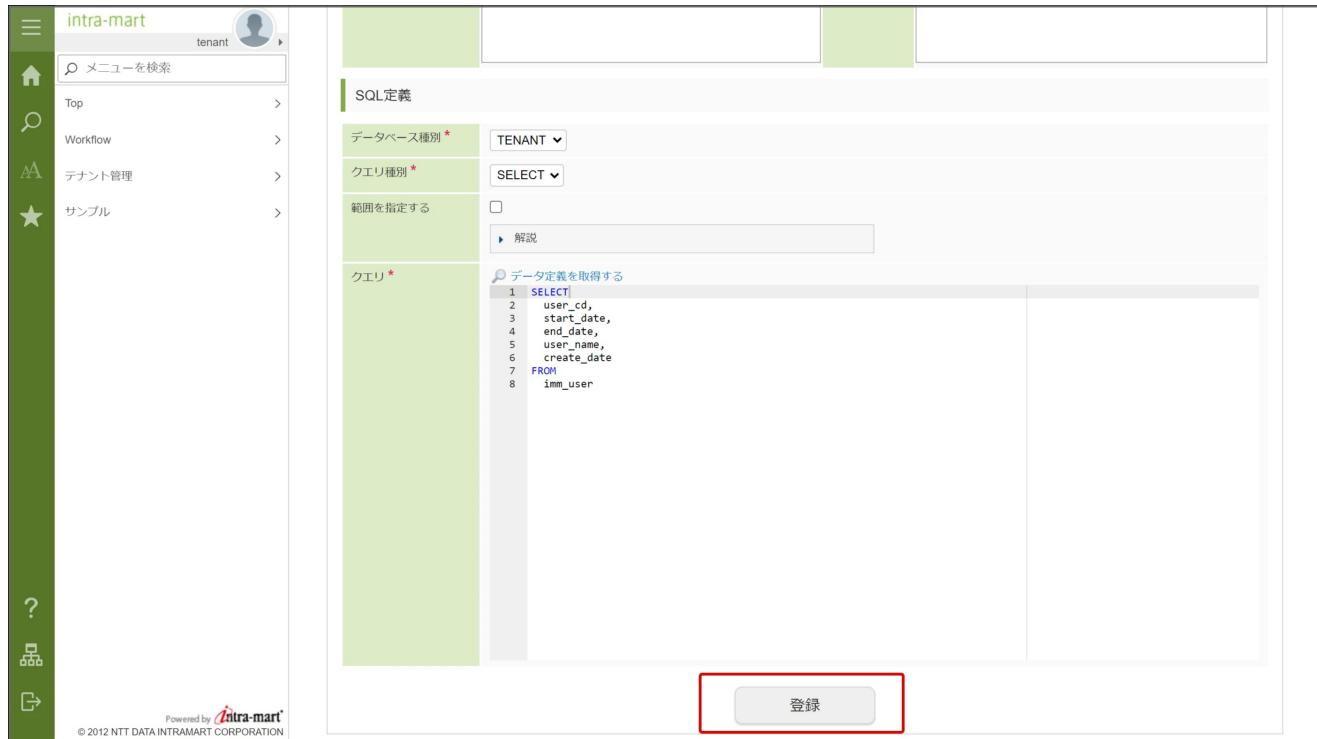
The screenshot shows the 'SQL Definition Editor' interface. On the left, there's a sidebar with navigation links like Home, Workflow, Tenant Management, and Samples. The main area has two tabs: 'Input Value' (入力値) and 'Return Value' (返却値). In the 'Input Value' tab, there are fields for boolean, string, integer, date, and object, along with a 'JSON Input' button. Below these are buttons for 'Add' (追加), 'Change Structure' (配下に配置する), and 'Delete' (削除). A 'Delete All' (全削除) button is also present. The 'Return Value' tab shows a similar structure for boolean, string, integer, date, and object, with a 'JSON Input' button. It also includes 'Add' (追加), 'Change Structure' (配下に配置する), and 'Delete' (削除) buttons, along with a 'Delete All' (全削除) button. A red box highlights the 'Return Value' section, specifically the expanded 'records <object[]>' section which lists user-related fields: user_cd, start_date, end_date, user_name, and create_date. At the bottom, there's a 'SQL Definition' (SQL定義) section with fields for 'Database Type' (データベース種別), 'Query Type' (クエリ種別), 'Scope' (範囲を指定する), and 'Query' (クエリ). A red box highlights the 'Query' field, which contains the SQL code:

```
1 SELECT
2   user_cd,
3   start_date,
4   end_date,
```

. Below the query, there's a 'Get Data Definition' (データ定義を取得する) button, which is also highlighted with a red box.

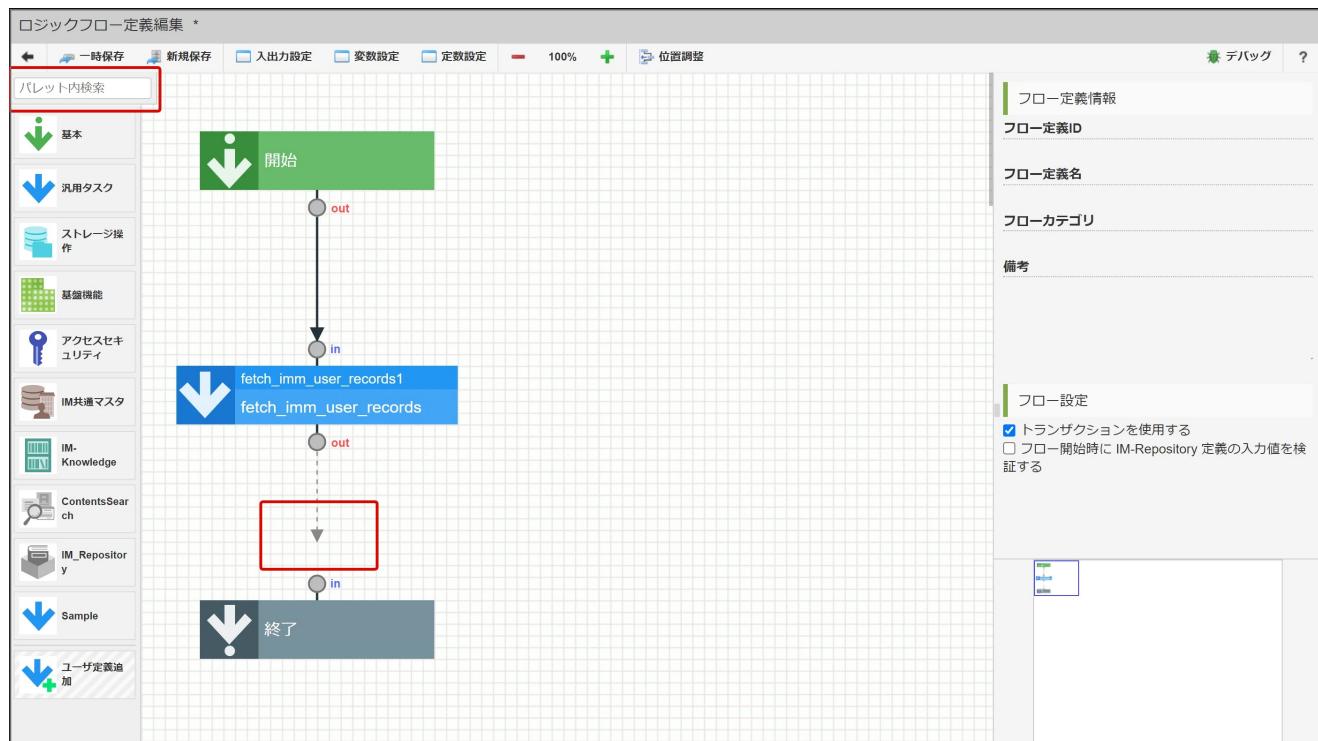
図：「SQL定義編集」画面 - 返却値の取得

5. 内容に問題がなければ、画面下部の「登録」をクリックして、SQL定義を保存します。



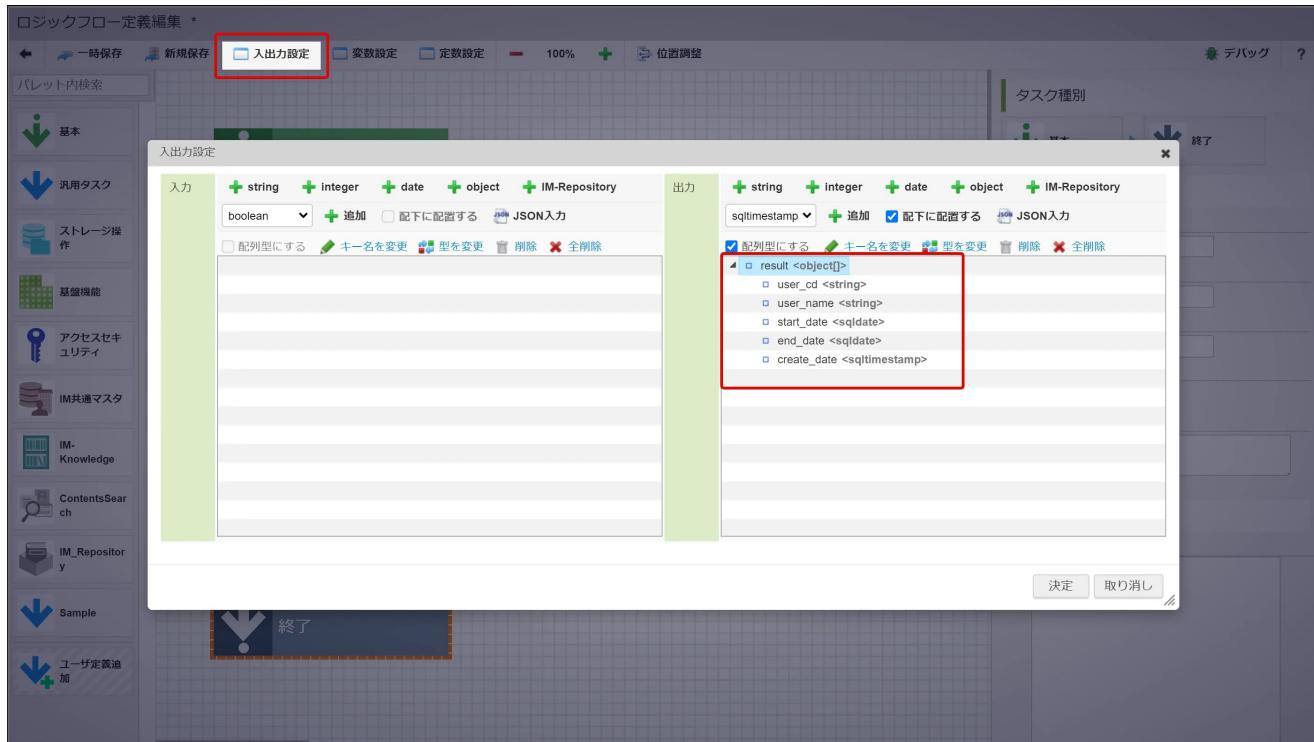
図：「SQL定義編集」画面 - SQL定義の保存

6. 「サイトマップ」 → 「LogicDesigner」 → 「フロー定義一覧」からロジックフローを新規作成し、「ロジックフロー定義編集」画面を表示します。
詳しい手順は「[ロジックフローの新規作成画面を表示する](#)」を参照してください。
7. 「開始タスク」、「fetch_imm_user_records (直前で作成したSQL定義)」「終了タスク」を配置し、順に接続します。
タスクはパレット上部の「パレット内検索」に名前を入力すると検索できます。



図：「ロジックフロー定義編集」画面 - タスクの接続

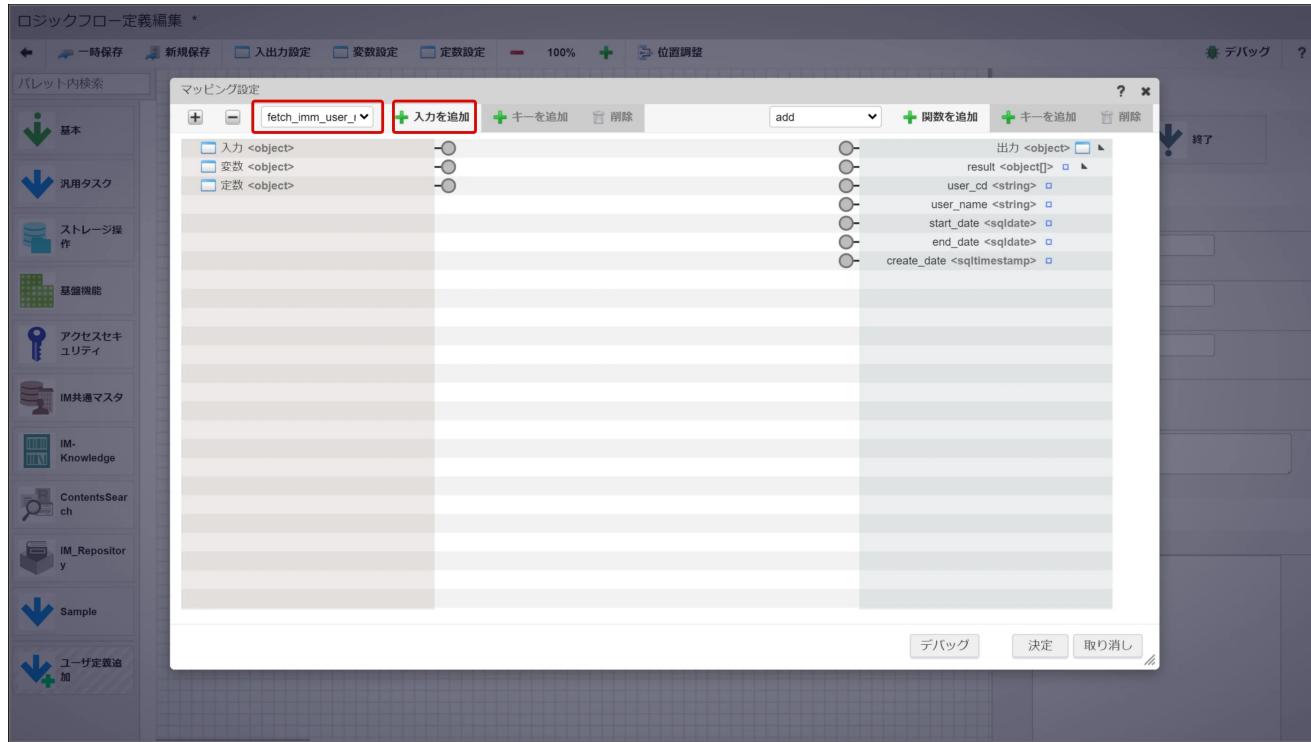
8. ヘッダの「入出力設定」をクリックし、出力値を以下のように設定し、「決定」をクリックします。
ロジックフローの出力結果をViewCreatorでリソースとして利用するには、指定された出力設定にする必要があります。
詳しくは「[ViewCreatorで利用可能なロジックフローの出力設定](#)」を参照してください。



図：「入出力設定」画面 - 出力の設定

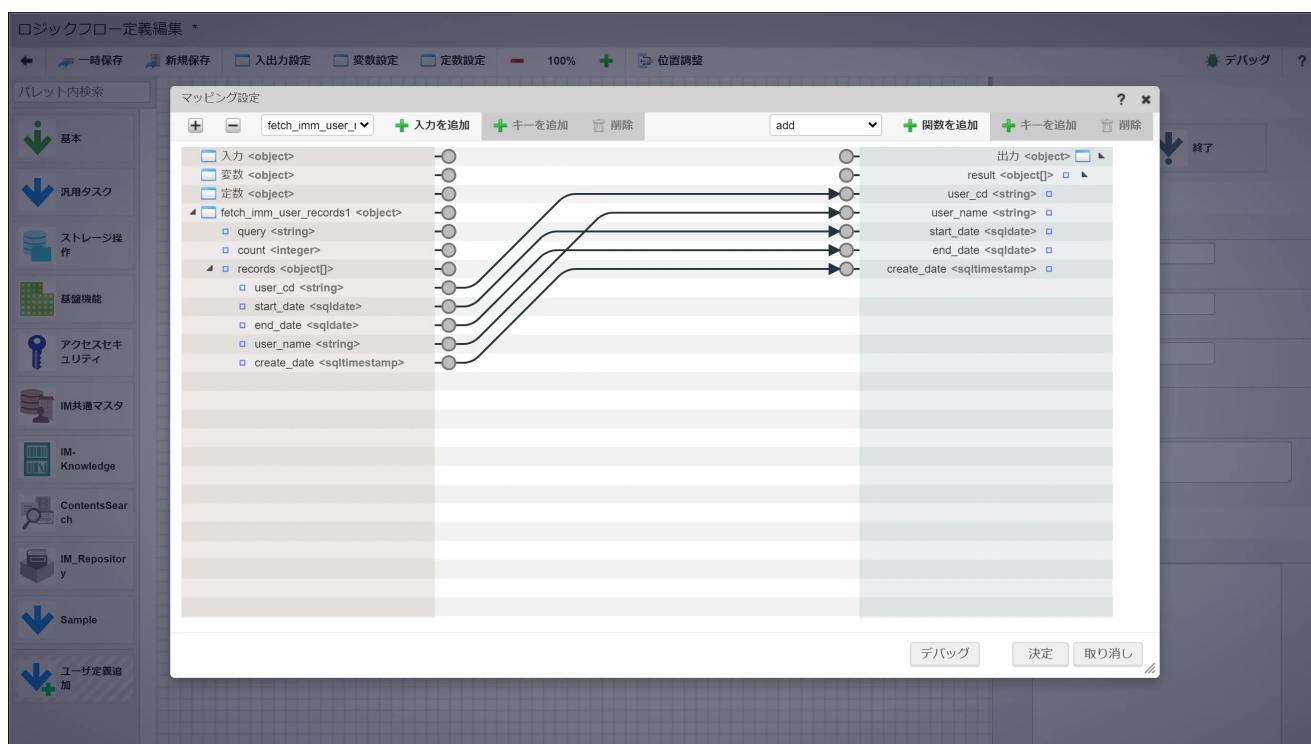
出力値	型
result	<object[]>
user_cd	<string> ※ result の配下に配置する
user_name	<string> ※ result の配下に配置する
start_date	<sqldate> ※ result の配下に配置する
end_date	<sqldate> ※ result の配下に配置する
create_date	<sqltimestamp> ※ result の配下に配置する

9. 「終了」タスクのマッピング設定を開き、ヘッダ内の左上に位置するセレクトボックスをクリックし、以下の項目を選択します。
 - エイリアス一覧 - fetch_imm_user_records1
10. セレクトボックスの中身が変更されたことを確認したら、右側にある「入力を追加」をクリックします。



図：「マッピング設定」画面 - 入力の追加

11. マッピング設定を以下のように設定し、「決定」をクリックします。



図：「マッピング設定」画面 - 「終了」タスクのマッピング設定

入力 (始点)	出力 (終点)
fetch_imm_user_records1 <object> └ records <object[]> └ user_cd <string>	出力 <object> └ result <object[]> └ user_cd <string>
fetch_imm_user_records1 <object> └ records <object[]> └ user_name <string>	出力 <object> └ result <object[]> └ user_name <string>
fetch_imm_user_records1 <object> └ records <object[]> └ start_date <sqldate>	出力 <object> └ result <object[]> └ start_date <sqldate>

入力（始点）

出力（終点）

fetch_imm_user_records1 <object>	出力 <object>
└ records <object[]>	└ result <object[]>
└ end_date <sqldate>	└ end_date <sqldate>
fetch_imm_user_records1 <object>	出力 <object>
└ records <object[]>	└ result <object[]>
└ create_date <sqltimestamp>	└ create_date <sqltimestamp>

12. 「新規保存」ボタンをクリックし、以下を設定してロジックフローを保存します。

- フロー定義ID : `fetch_imm_user_records_flow`
- フロー定義名 : `fetch_imm_user_records_flow`
- フローカテゴリ :
 - ID : 任意のものを選択してください。
 - 名称 : 任意のものを選択してください。

ロジックフローを ViewCreator で利用する

次に、ロジックフローをもとに ViewCreator のデータ参照を作成するための準備を行います。
具体的には、先ほど作成したロジックフローをリソースとして登録し、クエリを作成します。

ロジックフローを ViewCreator のリソースとして登録する

「ロジックフロー管理」画面で、リソースとして利用したいフローを登録します。

1. 「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。
2. フロー定義ID「`fetch_imm_user_records_flow`」の×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。

The screenshot shows the 'Logic Flow Management' page in the intra-mart Accel Platform. On the left is a navigation sidebar with links like 'Top', 'Workflow', 'Tenant Management', and 'Sample'. The main area is titled 'Logic Flow Management' and contains a table of logic flows. The first row, which has a checked checkbox in the 'ViewCreator 使用' column and a green checkmark in the 'フロー定義ID' column, is highlighted with a red box. This row represents the flow 'fetch_imm_user_records_flow' with the name 'fetch_imm_user_records_flow'. Other rows show various flows like 'im_getStorageInfo', 'sample-accounts', etc., each with a red 'X' or green checkmark in the 'ViewCreator 使用' column. At the bottom of the table, there are pagination controls and a message indicating 6 items displayed.

ViewCreatorで使用	フロー定義ID	フロー定義名	フローカテゴリ	プレビュー
✓	fetch_imm_user_records_flow	fetch_imm_user_records_flow	Sample	
✓	im_getStorageInfo	ファイル一覧取得	Storage	
×	sample-accounts	List of Accounts	Sample	
×	sample-backup-authz-data	Backup authz data files	Sample	
×	sample-im-topics-to-log	Read intra-mart atom feed	Sample	
×	sample-im_workflow-notice-to-user	IM-Workflow notice to user	Sample	

図：「ロジックフロー管理」画面 - リソース登録

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をクエリに追加します。

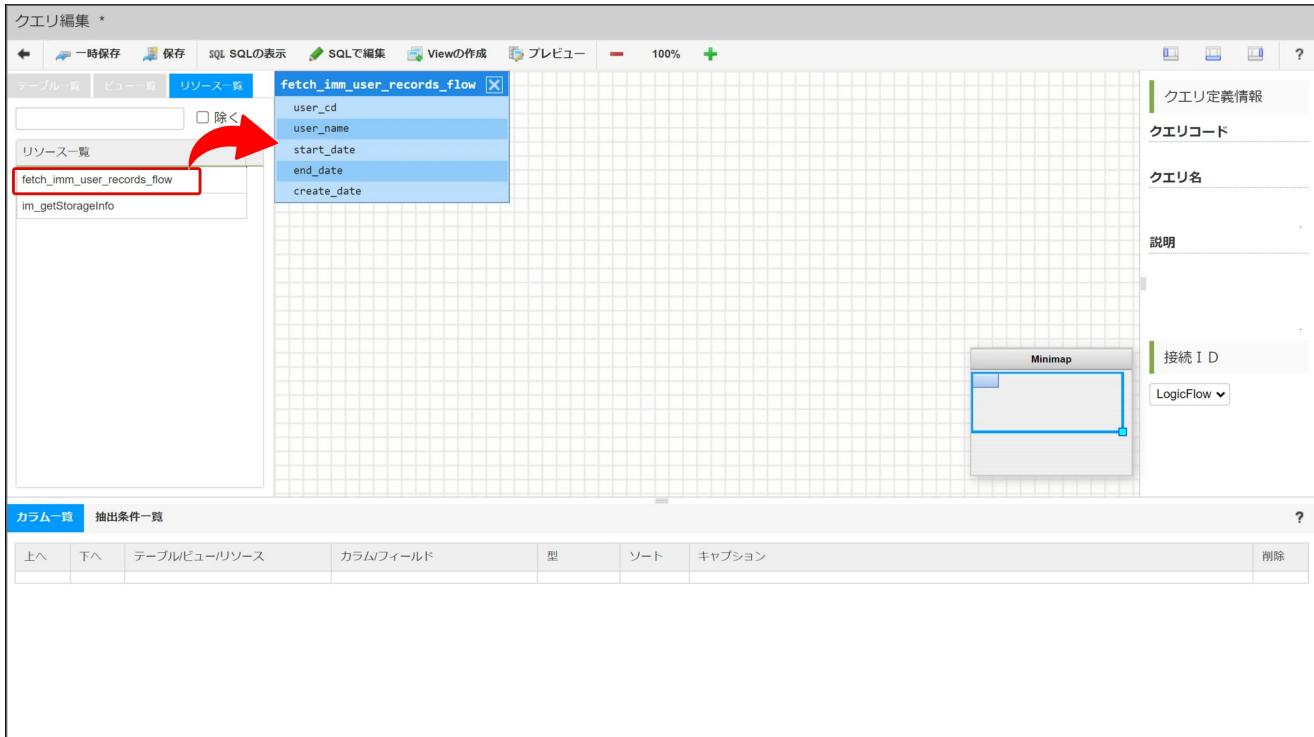
1. 「サイトマップ」→「ViewCreator」→「クエリー一覧」→「新規」をクリックし、「クエリ編集」画面を表示します。

図：「クエリ一覧」画面 - クエリ新規作成

2. 接続IDのプルダウンから「LogicFlow」を選択し、表示される確認ダイアログの「決定」ボタンをクリックします。

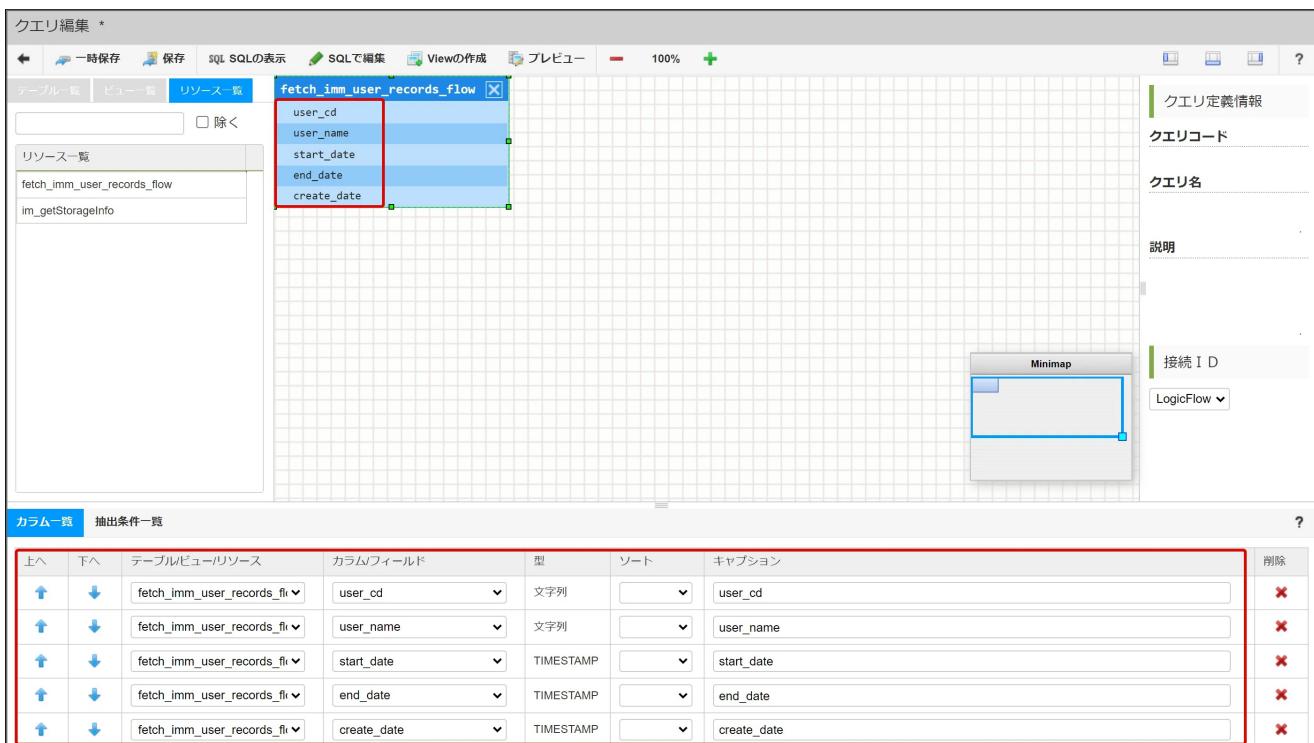
図：「クエリ編集」画面 - 接続IDの選択

3. リソース一覧に表示される「fetch_imm_user_records_flow」をダブルクリックし、テーブルウィンドウを表示します。



図：「クエリ編集」画面 - リソースの選択

4. 追加したリソースのテーブルウィンドウから以下のフィールドをダブルクリックし、クエリのカラムとして追加します。



図：「クエリ編集」画面 - カラム追加

- user_cd
- user_name
- start_date
- end_date
- create_date

5. 各カラムのキャプションを以下のように設定します。

キャプションを設定することで、データ参照画面で表示されるカラム名を変更できます。

カラム	フィールド	型	ソート	キャプション	
↑	fetch_imm_user_records.flr	user_cd	▼	ユーザコード	X
↑	fetch_imm_user_records.flr	user_name	▼	ユーザネーム	X
↑	fetch_imm_user_records.flr	start_date	▼	開始日	X
↑	fetch_imm_user_records.flr	end_date	▼	終了日	X
↑	fetch_imm_user_records.flr	create_date	▼	作成日時	X

図：「クエリ編集」画面 - キャプション設定

カラム/フィールド キャプション	
user_cd	ユーザコード
user_name	ユーザネーム
start_date	開始日
end_date	終了日
create_date	作成日時

6. 画面上部の「保存」ボタンをクリックし、以下を設定します。

設定ができたら、「決定」ボタンをクリックし、作成したクエリを登録します。

確認

クエリコード*	imm_user_query
クエリ名*	imm_user_query
説明	

決定
取り消し

カラム	フィールド	型	ソート	キャプション	
↑	fetch_imm_user_records.flr	user_cd	▼	ユーザコード	X
↑	fetch_imm_user_records.flr	user_name	▼	ユーザネーム	X
↑	fetch_imm_user_records.flr	start_date	▼	開始日	X
↑	fetch_imm_user_records.flr	end_date	▼	終了日	X
↑	fetch_imm_user_records.flr	create_date	▼	作成日時	X

図：「クエリ編集」画面 - クエリの保存

- クエリコード : imm_user_query

▪ クエリ名 : imm_user_query



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。

データ参照のリスト集計を作成する

ロジックフローをリソースとしたクエリが作成できたら、更にそれをを利用してデータ参照を作成します。今回は集計パターンとしてリスト集計を例に紹介します。

データ参照の種類の選択

「サイトマップ」→「ViewCreator」→「クエリ一覧」に遷移し、「クエリ名」が「imm_user_query」のリスト集計の アイコンをクリックして、「データ参照・新規作成」画面に遷移します。

The screenshot shows the 'Query List' screen in the ViewCreator application. On the left is a navigation sidebar with icons for Home, Search, Tenant Management, and Samples. The main area has a title bar 'クエリ一覧' with buttons for 'New', 'Copy', 'Delete', and 'Data Reference List'. Below is a search bar and a table listing data references. The first row, 'imm_user_query', has its 'Data Reference' icon highlighted with a red box. At the bottom are buttons for 'File Add...', 'Cancel', 'Delete', and 'Import'.

データ参照作成	クエリ名	クエリコード	接続 ID	更新日	作成者	エクスポート
	imm_user_query	imm_user_query	LogicFlow	2024/06/24 8:09	tenant	
	ViewCreatorの設定テーブルを利用した	5i7ur3opjhj2x	default	2013/11/05 18:40	tenant	
	日本のデータ	5i7o4c3kli1pj	default	2013/10/28 19:26	tenant	
	SQL 日本のデータ (データ結合)	8fhil6zaswe6qchu	default	2020/02/06 7:28	tenant	

図：「クエリ一覧」画面 - データ参照の新規作成

リスト集計の作成

1. 以下のように設定し、リスト集計を作成します。

The screenshot shows the 'Data Reference' configuration screen. Key fields highlighted with red boxes are:

- データ参照コード*: imm_user_list
- データ参照名*: imm_user_list

Other visible settings include:

- クエリ名: imm_user_query
- 集計パターン: リスト
- 説明: (empty)
- エクスポート機能の利用: 利用する (checked)

カテゴリ	項目名	設定値
データ参照設定	データ参照コード	imm_user_list
データ参照設定	データ参照	imm_user_list

The screenshot shows the 'Time Zone' and 'Column List' configuration screens.

Time Zone Settings:

- タイムゾーン設定の表示: タイムゾーン設定を表示する (checked)
- 画面表示のデフォルト・タイムゾーン: (デフォルト)ユーザが利用するタイムゾーン
- CSV出力のデフォルト・タイムゾーン: (デフォルト)(GMT+00:00) UTC

Column List Settings:

- タイムゾーン設定を表示する (checked)
- カラム一覧 (Table):

カラム	タイプ	表示	フォーマット	ソート順	パラメータ名	表示設定
ユーザコード(user_cd)	(dropdown)	<input checked="" type="checkbox"/>	(dropdown)	(dropdown)	(dropdown)	(dropdown)
ユーザネーム(user_name)	(dropdown)	<input checked="" type="checkbox"/>	(dropdown)	(dropdown)	(dropdown)	(dropdown)
開始日(start_date)	(dropdown)	<input checked="" type="checkbox"/>	(dropdown)	(dropdown)	(dropdown)	(dropdown)
終了日(end_date)	(dropdown)	<input checked="" type="checkbox"/>	(dropdown)	(dropdown)	(dropdown)	(dropdown)
作成日時(create_date)	(dropdown)	<input checked="" type="checkbox"/>	(dropdown)	(dropdown)	(dropdown)	(dropdown)

カテゴリ	項目名	設定値
タイムゾーン設定	タイムゾーン設定の表示	<ul style="list-style-type: none"> チェックボックスをオンにしてください。 「タイムゾーン設定を表示する」にチェックすることで、任意のタイムゾーンを選択し、選択したタイムゾーンで画面表示、CSV出力を行えます。 <p>注意事項等は「リスト集計の作成」を参照してください。</p>

カテゴリ	項目名	設定値
カラム一覧	カラム : <code>start_date</code> - フォーマット	<ul style="list-style-type: none"> 空欄にしてください。 <p>空欄にすることで個人設定の「日付の時刻の形式」で設定したアカウントごとのフォーマットでタイムスタンプ・日付型レコードを表示できます。フォーマットを設定した場合は、フォーマットが優先されます。</p> <p>フォーマットは、「yyyy/MM/dd HH:mm:ss」のような形式で設定してください。</p>
カラム一覧	カラム : <code>end_date</code> - フォーマット	<ul style="list-style-type: none"> 空欄にしてください。
カラム一覧	カラム : <code>create_date</code> - フォーマット	<ul style="list-style-type: none"> 空欄にしてください。



コラム

データ参照画面で表示されるレコードの絞り込みがしたい場合は、「[検索設定について](#)」を参照してください。

2. 画面下部の「登録して一覧へ戻る」ボタンをクリックし、作成したリスト集計を登録します。

データ参照画面のスクリーンショットです。左側にはナビゲーションメニューがあり、「Top」、「Workflow」、「テナント管理」、「サンプル」などのリンクがあります。右側には二つの富テキストエディタが表示されています。下部には「権限設定」というセクションがあり、「参照権」、「追加」、「ゲストユーザ」、「認証済みユーザ」のチェックボックスがあります。最下部には「登録して一覧へ戻る」ボタンがあります。

図：「データ参照・新規作成」画面 - データ参照の登録

作成したデータ参照を閲覧する

作成したデータ参照を閲覧します。

1. データ参照一覧で「imm_user_list」のリンクをクリックし、作成したデータ参照を閲覧します。

	データ参照名	更新日	作成者	エクスポート
<input type="checkbox"/>	2000年度版・地域 - 都道府県 サマリ集計	2013/11/05 0:18	tenant	
<input type="checkbox"/>	2000年度版・地域 - 都道府県 サマリ集計/面積順	2013/10/30 21:20	tenant	
<input type="checkbox"/>	2010年度版・地域 - 都道府県 サマリ集計	2020/02/06 7:47	tenant	
<input type="checkbox"/>	2010年度版・地域 - 都道府県 サマリ集計/面積順	2020/02/06 7:38	tenant	
<input type="checkbox"/>	サマリ集計から作成した棒グラフサンプル	2013/11/05 18:29	tenant	
<input type="checkbox"/>	データ参照一覧・サンプル1	2013/11/05 20:35	tenant	
<input type="checkbox"/>	データ参照一覧・サンプル2(クリエイティブでグループ化)	2013/11/05 20:35	tenant	
<input type="checkbox"/>	imm_user_list	2024/06/24 8:15	tenant	
<input type="checkbox"/>	年度・都道府県別人口 - クロス集計	2013/11/05 18:34	tenant	
<input type="checkbox"/>	関東の人口推移 (データ結合)	2020/02/06 8:15	tenant	
<input type="checkbox"/>	関東の人口推移 (データ結合) グラフサンプル	2020/02/06 8:07	tenant	
<input type="checkbox"/>	地域 - 都道府県 年度 グループ化リスト(人口)	2013/10/30 21:25	tenant	
<input type="checkbox"/>	都道府県別 面積トップ10・円グラフ	2013/11/05 0:12	tenant	
<input type="checkbox"/>	年度 - 地域 - 都道府県 グループ化リスト(人口)	2013/10/30 21:25	tenant	
<input type="checkbox"/>	年度 - 地域 - 都道府県 サマリ集計	2013/11/01 0:00	tenant	
<input type="checkbox"/>	年度 - 地域 - 都道府県 サマリ集計 (データ結合)	2020/02/06 7:41	tenant	

図：「データ参照一覧」画面 - データ参照の選択

2. imm_user のコードの一覧が表示されます。

imm_user_list					CSV出力	帳票出力	
	user code	user name	start date	end date	created date		
	tenant	tenant	1900/01/01 00:00:00	3000/01/01 00:00:00	2025/03/10 15:00:01		
	tenant	tenant	1900/01/01 00:00:00	3000/01/01 00:00:00	2025/03/10 15:00:01		
	tenant	tenant	1900/01/01 00:00:00	3000/01/01 00:00:00	2025/03/10 15:00:01		
	ueda	上田辰男	1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00		
	ueda	上田辰男	2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00		
	aoyagi	青柳辰巳	1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00		
	aoyagi	青柳辰巳	2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00		
	hayashi	林政義	1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00		
	hayashi	林政義	2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00		
	maruyama	円山益男	1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00		
	maruyama	円山益男	2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00		
	sekine	関根千香	1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00		
	sekine	関根千香	2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00		
	terada	寺田雅彦	1900/01/01 00:00:00	2000/01/01 00:00:00	2012/08/08 09:00:00		
	terada	寺田雅彦	2000/01/01 00:00:00	3000/01/01 00:00:00	2012/08/08 09:00:00		

図：データ参照画面

ViewCreator SQLビルダ アシスタント

ViewCreator SQLビルダ アシスタント の利用例

ここでは、実際にアシスタントを利用して、SQLを生成する例をご紹介します。

ユーザーを取得するSQLを生成する

以下のように、アシスタントに指示します。

ユーザの基本的な情報を一覧化して欲しい。

アシスタント の回答例です。

ユーザの基本的な情報を一覧化するためのSQLクエリは以下の通りです。このクエリでは、`imm_user`テーブルからユーザに関する各種情報を取得します。

```
```sql
SELECT
 u.user_cd AS user_code, -- ユーザコード
 u.locale_id AS locale_id, -- ロケールID
 u.term_cd AS term_code, -- 期間コード
 u.start_date AS start_date, -- 有効期間の開始日
 u.end_date AS end_date, -- 有効期間の終了日
 u.user_name AS user_name, -- ユーザ名
 u.user_search_name AS user_search_name, -- ユーザ検索名
 u.sex AS gender, -- 性別
 u.country_cd AS country_code, -- 国コード
 u.zip_code AS zip_code, -- 郵便番号
 u.address1 AS address1, -- 住所1
 u.address2 AS address2, -- 住所2
 u.address3 AS address3, -- 住所3
 u.telephone_number AS telephone, -- 電話番号
 u.extension_number AS extension_number, -- 内線番号
 u.fax_number AS fax_number, -- FAX番号
 u.extension_fax_number AS extension_fax_number, -- 内線FAX番号
 u.mobile_number AS mobile_number, -- 携帯電話番号
 u.email_address1 AS email1, -- メールアドレス1
 u.email_address2 AS email2, -- メールアドレス2
 u.mobile_email_address AS mobile_email, -- 携帯メールアドレス
 u.url AS url, -- URL
 u.notes AS notes, -- 備考
 u.delete_flag AS delete_flag, -- 削除フラグ
 u.sort_key AS sort_key, -- ソートキー
 u.create_user_cd AS created_by, -- 作成者
 u.create_date AS created_at, -- 作成日
 u.record_user_cd AS updated_by, -- 最終更新者
 u.record_date AS updated_at -- 最終更新日
FROM
 imm_user u -- ユーザーテーブル
WHERE
 u.locale_id = 'ja' -- ロケールIDが'ja'のユーザ
 AND u.delete_flag = '0'; -- ユーザが削除されていない
```

```

このクエリは、ロケールIDが'ja'であり、削除フラグが'0'（有効）のユーザの情報を一覧で取得します。

項目数が多いので、代表的な項目に絞るように続けて指示します。

代表的な項目のみに絞ってください。

アシスタント の回答例です。

以下のSQLクエリは、ユーザの代表的な情報をのみを一覧化するためのものです。ユーザコード、ロケールID、ユーザ名、およびメールアドレス1と電話番号に絞っています。

```
```sql
SELECT
 imm_user.user_cd AS user_code,
 imm_user.locale_id AS locale_id,
 imm_user.user_name AS user_name,
 imm_user.email_address1 AS email_address1,
 imm_user.telephone_number AS telephone_number
FROM
 imm_user
WHERE
 imm_user.locale_id = 'ja' -- ロケールIDが'ja'のユーザ
 AND imm_user.delete_flag = '0' -- ユーザが削除されていない
ORDER BY
 imm_user.user_cd;
```
```

このクエリは、必要な項目を厳選しており、ユーザコード順に結果を整列させます。

各ユーザが持っているロールも一覧で見たいので、続けて指示します。

各ユーザが保持しているロールの情報も取得してください。

アシスタントの回答例です。

以下のSQLクエリは、ユーザの基本的な情報に加えて、各ユーザが保持しているロールの情報を取得します。`imm_user`テーブルと`b_m_account_role_b`テーブルを結合し、ユーザコード、ユーザ名、メールアドレス、電話番号、ロールID、およびロール名を取得しています。

```
```sql
SELECT
 u.user_cd AS user_code,
 u.locale_id AS locale_id,
 u.user_name AS user_name,
 u.email_address1 AS email_address1,
 u.telephone_number AS telephone_number,
 r.role_id AS role_id,
 r.role_name AS role_name
FROM
 imm_user u
JOIN
 b_m_account_role_b ar ON u.user_cd = ar.user_cd
JOIN
 b_m_role_b r ON ar.role_id = r.role_id
WHERE
 u.locale_id = 'ja' -- ロケールIDが'ja'のユーザ
 AND u.delete_flag = '0' -- ユーザが削除されていない
 AND ar.valid_start_date <= CURRENT_TIMESTAMP -- ロールの有効開始日
 AND ar.valid_end_date >= CURRENT_TIMESTAMP -- ロールの有効終了日
ORDER BY
 u.user_cd, r.role_id;
```
```

このクエリでは、ユーザとそのロールに関する必要な情報を一度の実行で取得し、ユーザコードおよびロールID順に整列させています。ロールの有効期間も考慮されています。

このように、AIと自然文で対話しながら、SQLを生成できます。

実行できないSQLや意図しない結果を取得するSQLだった場合は、エラーコードやエラーメッセージなどのできるだけ具体的な情報を追加で提供すると適切な回答が得られる可能性が高まります。

SQL生成に利用するテーブルを予め指定する場合

利用して欲しいテーブルが最初から分かっている場合は、指示に含めておくと、より適切なSQLが生成される可能性が高まります。

指示

ユーザの基本的な情報を一覧化してください。
一覧には、ユーザが保持しているロールの情報も含めてください。

条件

次のテーブルを参照してください。

imm_user, b_m_account_b, b_m_account_role_b



コラム

上記は、マークダウントラックを利用した指示の例です。

アシスタントの回答例です。

以下は、ユーザの基本情報およびその保持するロールの情報を一覧化するためのSQLクエリです。

```
```sql
SELECT
 u.user_cd AS user_code, -- ユーザコード
 u.user_name AS user_name, -- ユーザ名
 u.locale_id AS user_locale_id, -- ロケールID
 u.term_cd AS user_term_code, -- 期間コード
 u.start_date AS user_start_date, -- ユーザの有効期間の開始日
 u.end_date AS user_end_date, -- ユーザの有効期間の終了日
 u.sex AS user_sex, -- 性別
 u.country_cd AS user_country_code, -- 国コード
 a.first_day_of_week AS user_first_day_of_week, -- 週の開始曜日
 ar.role_id AS role_id, -- ロールID
 ar.valid_start_date AS role_valid_start_date, -- ロール有効開始日
 ar.valid_end_date AS role_valid_end_date -- ロール有効終了日
FROM
 imm_user u -- ユーザ情報を管理するテーブル
LEFT JOIN
 b_m_account_b a ON u.user_cd = a.user_cd -- アカウント情報を管理するテーブルとの結合
LEFT JOIN
 b_m_account_role_b ar ON u.user_cd = ar.user_cd -- アカウントに付与されたロール情報を管理するテーブルとの結合
WHERE
 u.locale_id = 'ja' -- 日本語のロケールIDを持つレコードのみを抽出
 AND u.delete_flag = '0'; -- 削除されていないユーザのみを対象
```

```

このクエリは、`imm_user`、`b_m_account_b`、および`b_m_account_role_b`テーブルを結合し、ユーザの基本情報とそのロールの情報を取得します。データは、日本語ロケールIDを持つ有効なユーザのみにフィルタリングされています。

どのようなテーブルがあるのか知りたい場合

知りたいテーブルに関する指示をします。

このシステムで利用可能なユーザに関するテーブルの一覧を出してください。

アシスタントの回答例です。

ユーザに関するテーブルは以下の通りです。

1. **ユーザ**

- テーブル名: `imm_user`
- 説明: ユーザの個人情報のマスタデータを管理するテーブルです。

2. **ユーザ分類**

- テーブル名: `imm_user_ctg`
- 説明: ユーザ分類のマスタデータを管理するテーブルです。

3. **ユーザ分類所属**

- テーブル名: `imm_user_ctg_ath`
- 説明: ユーザがユーザ分類項目に分類されていることを表すマスタデータを管理するテーブルです。

4. **ユーザ分類項目**

- テーブル名: `imm_user_ctg_itm`
- 説明: ユーザ分類ごとに定義されている詳細（マスタデータ）を管理します。

5. **ユーザプロファイル画像**

- テーブル名: `imm_user_image`
- 説明: ユーザのプロファイル画像を管理するテーブルです。

これらのテーブルがユーザに関するデータを管理しています。他に知りたい情報がありましたら、教えてください。

imm_user について詳細を知りたいので、続けて指示します。

imm_user について詳細を知りたいです。

アシスタント の回答例です。

テーブル `imm_user` に関する詳細は以下の通りです。

```
### テーブル名: imm_user
- **論理名**: ユーザ
- **説明**: ユーザの個人情報のマスタデータを管理するテーブルです。
```

カラム情報

| 物理名 | 論理名 | 型 | nullable | 説明 |
|----------------------|--------------|-----------|----------|-------------------------|
| user_cd | ユーザコード | varchar | false | |
| locale_id | ロケールID | varchar | false | |
| term_cd | 期間コード | varchar | false | ユーザの期間に対して一意となるコード |
| start_date | ユーザの有効期間の開始日 | date | false | |
| end_date | ユーザの有効期間の終了日 | date | false | |
| user_name | ユーザ名 | varchar | false | |
| user_search_name | ユーザ検索名 | varchar | true | |
| sex | 性別 | varchar | true | 0:男性 1:女性 2:その他 9:回答しない |
| country_cd | 国コード | varchar | true | ISO 3166-1 |
| zip_code | 郵便番号 | varchar | true | |
| address1 | 住所1 | varchar | true | |
| address2 | 住所2 | varchar | true | |
| address3 | 住所3 | varchar | true | |
| telephone_number | 電話番号 | varchar | true | |
| extension_number | 内線番号 | varchar | true | |
| fax_number | FAX番号 | varchar | true | |
| extension_fax_number | 内線FAX番号 | varchar | true | |
| mobile_number | 携帯電話番号 | varchar | true | |
| email_address1 | メールアドレス1 | varchar | true | |
| email_address2 | メールアドレス2 | varchar | true | |
| mobile_email_address | 携帯メールアドレス | varchar | true | |
| url | URL | varchar | true | |
| notes | 備考 | varchar | true | |
| delete_flag | 削除フラグ | varchar | false | 0:有効 1:無効 |
| sort_key | ソートキー | numeric | false | |
| create_user_cd | 作成者 | varchar | false | |
| create_date | 作成日 | timestamp | false | |
| record_user_cd | 最終更新者 | varchar | false | |
| record_date | 最終更新日 | timestamp | false | |

このテーブルでは、各ユーザの基本的な情報や連絡先、住所、そして削除フラグなどの情報が管理されています。他に知りたいことがあれば教えてください。

独自のテーブルメタデータ情報を追加する

アシスタントによるSQL生成では、テーブルメタデータ情報（テーブル・キャプション情報、サンプルSQL情報、テーブルカテゴリ情報）を利用しています。

お客様が独自に利用しているテーブルのテーブルメタデータ情報を追加するには、下記の手順に従ってテーブルメタデータ情報を用意し、ジョブネットを実行してください。



アシスタント機能は 2024 Autumn(Jasmine) から利用可能です。

2024 Autumn(Jasmine) から製品標準で提供しているテーブルのテーブルメタデータ情報（テーブル・キャプション情報、サンプルSQL情報、テーブルカテゴリ情報）はテナント環境セットアップ時に登録されます。

目次

- テーブルメタデータ情報の用意
 - テーブル・キャプション情報
 - 「テーブル・キャプション登録」画面を利用する
 - 画面から登録する
 - インポートする
 - ジョブネットを利用する
 - サンプルSQL情報
 - テーブルカテゴリ情報
 - ジョブネットの実行

テーブルメタデータ情報の用意

テーブルメタデータ情報（テーブル・キャプション情報、サンプルSQL情報、テーブルカテゴリ情報）について説明します。

テーブル・キャプション情報

テーブル、および、フィールドに対するキャプションとコメントです。



注意

アシスタントは日本語ロケールの情報を利用しています。

日本語以外を利用している場合でも、テーブル・キャプション情報の日本語ロケールへ登録してください。

※登録内容は日本語である必要はありません。

「テーブル・キャプション登録」画面を利用する

テーブル・キャプション情報は「テーブル・キャプション登録」画面を利用して登録できます。

画面から登録する

「テーブル・キャプション登録」画面から登録してください。

詳細は、「TableMaintenance 管理者操作ガイド」 - 「テーブルの拡張情報の設定」を参照してください。

インポートする

「テーブル・キャプション登録」画面のインポート機能を利用して登録してください。

インポートの場合、XML定義ファイルを利用することで複数のテーブルに対して一括で登録できます。

詳細は、「TableMaintenance 管理者操作ガイド」 - 「インポートとエクスポート」を参照してください。

ジョブネットを利用する

ジョブネットを利用してテーブル・キャプション情報を登録できます。

ジョブネットを利用する場合は、XML定義ファイルを用意する必要があります。

用意するXML定義ファイルのフォーマットは「TableMaintenance 管理者操作ガイド」 - 「インポートとエクスポート」で扱うXML定義ファイルと同様のフォーマットです。

XML定義ファイルは下記の命名規則に従ったファイル名で作成してください。

- table_definition_xxx.xml
※ xxxの部分は任意
(例) table_definition_viewcreator.xml

XML定義ファイルを用意した後は「ジョブネットの実行」から登録してください。

サンプルSQL情報

テーブルを使用したSQL例です。

サンプルSQL情報のフォーマットは下記です。

```
-- SQLクエリの説明1
-- SQLクエリの説明1
```

SELECT**...****FROM****...****WHERE****...****ORDER BY** ...**;**

```
-- SQLクエリの説明2
-- SQLクエリの説明2
```

SELECT**...****FROM****...****WHERE****...****ORDER BY** ...**;**

サンプルSQL情報を作成する上での注意点は下記です。

- SQLクエリには必ずコメントで説明を記載してください。
- SQLの最後は ; (セミコロン) で終えて、改行してください。
- 複数のサンプルクエリを記載する場合は、クエリ間に空白行を入れてください。
- プリペアドステートメントは使用せずに、データベース操作などでそのまま実行可能なSQLを用意してください。
 - データベース操作については、「[システム管理者操作ガイド](#)」 - 「[データベース操作](#)」を参照してください。
- 利用しているデータベースで実行可能なSQLを用意してください。

サンプルSQL情報は下記の命名規則に従ったファイル名で作成してください。

- example_query_xxx.sql
※ xxxの部分は任意
(例) example_query_viewcreator.sql

テーブルカテゴリ情報

各テーブルを分類する情報です。

テーブルカテゴリ情報のフォーマットは下記です。

```
{
  "summary": "カテゴリの要約文を書いてください",
  "description": "カテゴリの詳細説明文を書いてください",
  "tables": [
    {
      "name": "テーブル物理名"
    },
    {
      "name": "テーブル物理名にマッチする正規表現"
    }
  ],
  "categories": [
    {
      "summary": "サブカテゴリの要約文を書いてください",
      "description": "サブカテゴリの詳細説明文を書いてください",
      "tables": [],
      "categories": []
    }
  ]
}
```

テーブル物理名の記載方法については正規表現も利用できます。

- (例) sample_table_1, sample_table_2をまとめて指定する場合

```
"tables": [
  {
    "name": "sample_table_.*"
  }
]
```

カテゴリは階層構造で記述できます。

テーブルカテゴリ情報は下記の命名規則に従ったファイル名で作成してください。

- table_category_xxx.json
※ xxxの部分は任意
(例) table_category_viewcreator.json

ジョブネットの実行

各テーブルメタデータ情報は下記のジョブネットから登録できます。

- テナントデータベース上のテーブルのテーブルメタデータ情報を登録する場合
 - 「ジョブ・ジョブネットリファレンス」 - 「[テーブルメタデータ同期+ベクトルデータベース登録](#)」
- シェアードデータベース上のテーブルのテーブルメタデータ情報を登録する場合
 - 「ジョブ・ジョブネットリファレンス」 - 「[テーブルメタデータ同期+ベクトルデータベース登録（シェアードデータベース）](#)」

「[テーブルメタデータ情報の用意](#)」で用意した各テーブルメタデータ情報のファイルを、ストレージパスに設定したストレージの直下に配置してください。

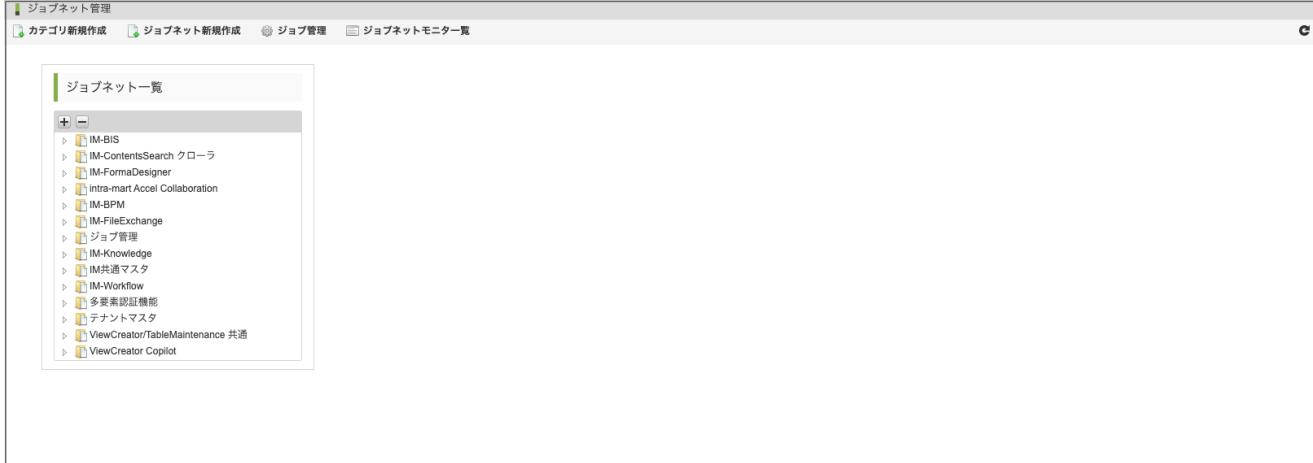
ストレージパスがデフォルト値「vc_tm_common/tablemetadata」の場合のファイル配置の例は下記です。

- vc_tm_common
 - └ tablemetadata
 - ├ table_definition_xxx.xml
 - ├ table_category_xxx.json
 - └ example_query_xxx.sql

※ xxxの部分は任意

登録したいテーブルメタデータ情報を配置後、下記の手順でジョブネットを実行してください。

1. 「サイトマップ」 → 「テナント管理」 → 「ジョブ管理」 → 「ジョブネット管理」から、「ジョブネット管理」画面を表示します。



図：「ジョブネット管理」画面

2. 「ViewCreator Copilot」のカテゴリをクリックします。

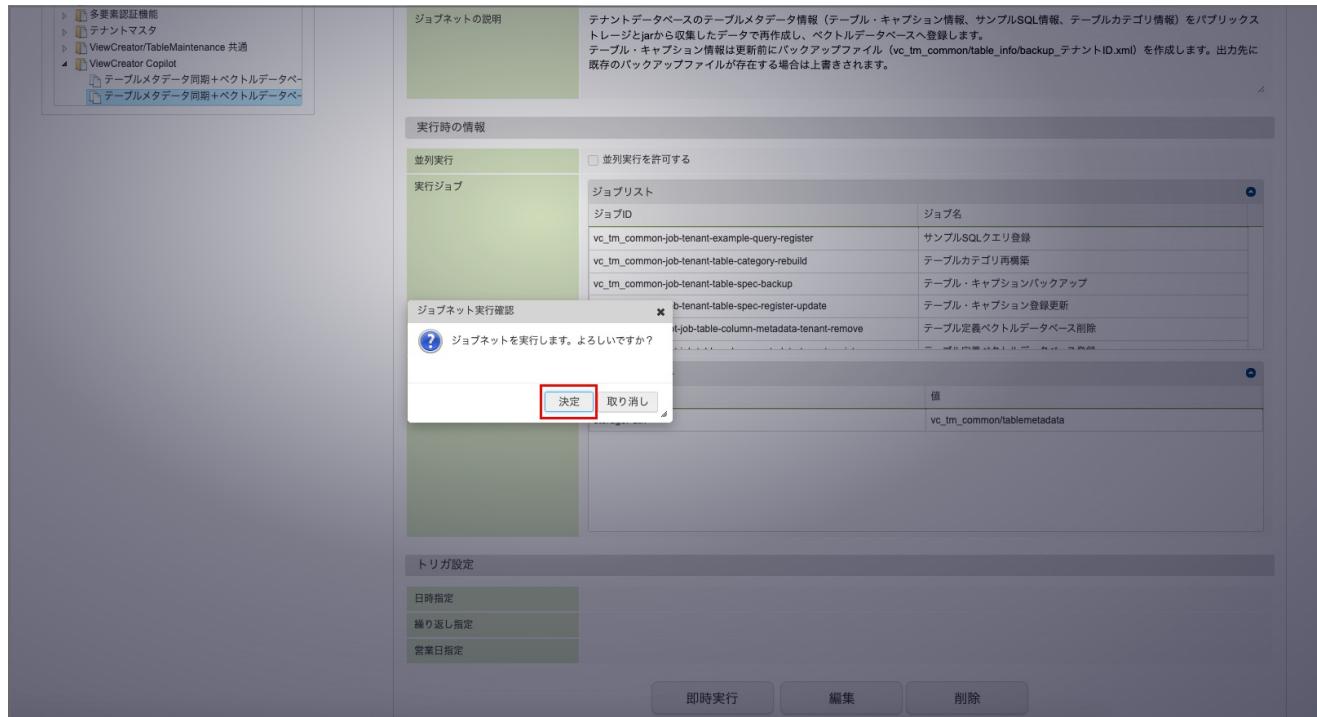


図：「ジョブネット管理」画面 - 「ViewCreator Copilot」カテゴリのジョブネット選択

3. 取り込むテーブルメタデータ情報に合わせたジョブネットをクリックし、必要に応じて実行パラメータを変更の上、即時実行をクリックします。



図：「ジョブネット管理」画面 - 実行パラメータ確認



図：「ジョブネット管理」画面 - 即時実行クリック

4. 「ジョブネットモニター覧」画面から実行したジョブネットの結果を確認してください。

注意

製品標準で提供しているテーブルに対してテーブル・キャプション情報を独自に登録している場合

ジョブネットを実行すると、製品標準で提供しているテーブルに対して、製品標準のテーブル・キャプション情報に上書きされます。

パブリックストレージ「vc_tm_common/table_info」直下にバックアップファイルが作成されるため、必要に応じて差分を「テーブル・キャプション登録」画面から再登録してください。

ViewCreator SQLビルダ アシスタント 利用時のコスト

ViewCreator SQLビルダ アシスタント を利用する際に発生する利用トークンについて説明します。

トークンとは、AIが文章を処理する際の最小単位（単語や記号など）です。利用料金はこのトークン数に応じて決まります。

コラム

利用トークン数について

このドキュメントに記載されている利用トークン数は一例です。

チャットモデル、質問の複雑さ、言語等によって変動します。

なお、記載されている内容は 2025 Autumn(Lilac) リリース時点のものです。利用サービス、モデル、仕様変更等によって大きく変動する可能性があります。

注意

生成AIサービスの利用料金について

実際請求される料金は各生成AIサービスの利用料金に準じます。

生成AIサービスの利用料金については、各サービス提供元にお問い合わせください。

ViewCreator SQLビルダ アシスタント の実行ステップ

ViewCreator SQLビルダ アシスタント では、AIが複数の処理を組み合わせて最適なSQLを生成する方式を採用しています。

質問の内容によっては処理を複数回実行することがあるため、一回の利用トークン数は変動します。

また、同じ質問であっても、AIサービスの特性（利用サービス、モデル等）により毎回トークン数が変動します。

各実行ステップとその役割は以下の通りです。

1. テーブル検索

- ユーザの質問に関連するデータベーステーブルを探します
- 質問から必要なデータが格納されているテーブルを自動で判断します

2. テーブル精度検証

- 見つかったテーブルが質問に対して適切かどうかを詳細に確認します
- テーブルの構造（カラム名、データ型、制約等）を取得・分析します
- より正確なSQL作成のため、このステップは複数回実行される場合があります

3. SQL生成

- 検証されたテーブル情報を元に、SQLクエリを作成します



コラム

効率的な利用方法

質問に具体的なテーブル名やカラム名を含めることで、テーブル検索やテーブル精度検証の処理負荷を軽減できます。これにより、利用トークン数を抑えてコストを削減できる可能性があります。

ViewCreator SQLビルダ アシスタント の利用トークン数の例

質問とトークン数の関係

質問の内容や複雑さによって、利用トークン数は大きく変動します。以下は標準的な質問の利用トークン数の例です。

以下のプロンプトを1回実行した場合

組織の一覧を出してください。

実行結果

| 生成AIサービス | チャットモデル名 | 入力トークン数 | 出力トークン数 | 実行ステップ内訳 |
|----------------------|--|---------|---------|---|
| OpenAI | gpt-4o-mini | 79,883 | 993 | テーブル検索 入力:4,674 出力:82
テーブル精度検証（1回目） 入力:24,100 出力:337
テーブル精度検証（2回目） 入力:41,574 出力:126
SQL生成 入力:9,535 出力:448
合計 入力:79,883 出力:993 |
| Azure OpenAI Service | gpt-4o-mini | 74,314 | 1,239 | テーブル検索 入力:4,695 出力:84
テーブル精度検証（1回目） 入力:24,143 出力:548
テーブル精度検証（2回目） 入力:24,137 出力:354
SQL生成 入力:21,339 出力:253
合計 入力:74,314 出力:1,239 |
| Amazon Bedrock | anthropic.claude-3-haiku-20240307-v1:0 | 75,368 | 1,235 | テーブル検索 入力:9,244 出力:165
テーブル精度検証（1回目） 入力:14,010 出力:343
テーブル精度検証（2回目） 入力:36,995 出力:156
SQL生成 入力:15,119 出力:571
合計 入力:75,368 出力:1,235 |

ログ関連

ViewCreator 操作ログについて

ViewCreator 操作ログでは「データ参照」画面を表示する等の操作で、データの取得が行われたことを以下のパスのログに記録します。

- WEB-INF/log/product/viewcreator/viewcreator-operation.log

ログの出力例は以下の通りです。

```
2025-05-28 04:05:30.637 resin-port-8080-88 INFO 5je7o5i6x1pcq 8hmgv0hy4a3g380 tenant 5i7o4o3kli1pj EXECUTE QUERY
2 0 2 default {drilldownParameters=[], searchParameters=[], sortParameters=[], startRow=1, length=300} 処理が正常に完了しました。
2025-05-28 04:06:38.912 resin-port-8080-106 INFO 5je7o5i98lch7 8hmgv1ynka3p780 tenant 5i7o4o3kli1pj sample_japan_list
EXECUTE LIST_DATAVIEW 2 1 1 default {drilldownParameters=[], searchParameters=[{columnCode=ccd00003, keyword=島}], sortParameters=[{columnCode=ccd00002, columnName=, ascending=false}], startRow=51, length=0} 処理が正常に完了しました。
2025-05-28 04:06:44.051 resin-port-8080-106 INFO 5je7o5i9ck3hb 8hmgv22lda3pa80 tenant 5i7o4o3kli1pj sample_japan_list
EXPORT LIST_DATAVIEW 8 0 0 default {drilldownParameters=[], searchParameters=[{columnCode=ccd00003, keyword=島}], sortParameters=[{columnCode=ccd00002, columnName=, ascending=false}], startRow=0, length=0} 処理が正常に完了しました。
```

データ参照を利用してデータ取得を行ったユーザのユーザコードやデータ参照コード、SQLの実行に要した時間等が output されます。上記の例では、クエリ（クエリコード : 5i7o4o3kli1pj）の実行（プレビュー）や、リスト集計（データ参照コード : sample_japan_list）の実行、エクスポートが tenant ユーザによって行われ、それぞれ、2ミリ秒、2ミリ秒、8ミリ秒で完了したことがわかります。ログの出力先やレイアウトに関する設定は、「[ログ仕様書](#)」 - 「[ViewCreator 操作ログ](#)」を参照してください。

SQLの実行時間が閾値を超えていている場合

SQLの実行時間が閾値を超えていない場合は、INFO レベルのログが出力されますが、超えている場合は、以下のようにWARNレベルで出力されます。

閾値の設定については、「[設定ファイルリファレンス](#)」 - 「[ViewCreatorの設定](#)」の「ログ設定」を参照してください。

```
2025-05-28 04:06:44.051 resin-port-8080-106 WARN 5je7o5i9ck3hb 8hmgv22lda3pa80 tenant 5i7o4o3kli1pj
sample_japan_list EXECUTE LIST_DATAVIEW 32573 14433 13822 default {drilldownParameters=[], searchParameters=[{columnCode=ccd00003, keyword=島}], sortParameters=[{columnCode=ccd00002, columnName=, ascending=false}], startRow=0, length=0} 件数取得処理が閾値を超過しました。 閾値:5000ms 実際:14433ms
```

上記の例では、データ参照コード sample_japan_list の実行において、総件数取得SQL実行時間が閾値（5000ミリ秒）を超過し、実際の実行時間は14433ミリ秒であったことがわかります。

WARNレベルのログが出力された場合は、以下のように実行されたSQL文をメッセージに含む例外ログが出力されます。

```
log.generating.time=Wed May 28 04:44:25 UTC 2025
log.level=WARN
log.logger.name=VIEWCREATOR_OPERATION_LOG
log.id=5je7o5kie2grx
log.thread.id=resin-port-8080-105
log.thread.group=main
log.message=件数取得処理が閾値を超過しました。 閾値:1ms 実際:4ms

jp.co.intra_mart.system.viewcreator.log.exception.SlowSQLException: Execution time exceeded threshold. Execution time:
14433ms
SQL: SELECT COUNT(*) AS CNT FROM (SELECT 'x' AS im_vc_dummy FROM ( SELECT
    sample_population.year AS ccd00001,
    sample_region.name AS ccd00002,
    sample_prefecture.name AS ccd00003,
    sample_age.age AS ccd00004,
    sample_prefecture_area.area AS ccd00005,
    sample_population.population AS ccd00006

FROM
    sample_population
    LEFT OUTER JOIN sample_age
        ON sample_population.age_id = sample_age.id
    LEFT OUTER JOIN sample_prefecture
        ON sample_population.prefecture_id = sample_prefecture.id
    LEFT OUTER JOIN sample_prefecture_area
        ON sample_prefecture.id = sample_prefecture_area.id
    LEFT OUTER JOIN sample_region
        ON sample_prefecture.region_id = sample_region.id
) IM_QUERY_ ) VC_IM_QUERY_im_a
at
jp.co.intra_mart.system.viewcreator.log.ViewCreatorOperationTracer$1.onCountRetrievalExceeded(ViewCreatorOperationTracer.java:220
at
jp.co.intra_mart.system.viewcreator.log.StandardDurationThresholdChecker.lambda$notifyCountRetrievalCompleted$0(DurationThresholdChecke
at java.base/java.util.ArrayList.forEach(ArrayList.java:1541)
at
jp.co.intra_mart.system.viewcreator.log.StandardDurationThresholdChecker.notifyCountRetrievalCompleted(DurationThresholdChecker.ja
at jp.co.intra_mart.system.viewcreator.log.ViewCreatorOperationTracer.endCountRetrieval(ViewCreatorOperationTracer.java:147)
at jp.co.intra_mart.foundation.viewcreator.ExtendDataViewManager.getListResult(ExtendDataViewManager.java:1304)
at jp.co.intra_mart.foundation.viewcreator.DataViewManager$3.execute(DataViewManager.java:470)
```



コラム

閾値を超えたときだけログ出力を行いたい場合は、「[ログ仕様書](#)」 - 「[ViewCreator 操作ログ](#)」の「標準出力設定」で、ログレベルを WARN に設定してください。