

Apache Superset 調査報告書

1.00 版

2022 年 2 月 8 日



Contents

| | |
|------------------------------------|-----------|
| 1 概要 | 3 |
| 1.1 本書について | 3 |
| 1.2 Apache Superset について | 3 |
| 1.3 機能概観 | 3 |
| 2 対応データベース一覧 | 5 |
| 3 Apache Superset のインストール方法 | 7 |
| 3.1 環境 | 7 |
| 3.2 依存関係ソフトウェアのインストール | 7 |
| 3.3 Python Virtual Environment の設定 | 7 |
| 3.4 MariaDB のインストール | 8 |
| 3.5 Apache Superset のインストール | 8 |
| 3.6 Apache Superset の設定ファイルの作成 | 8 |
| 3.7 データベースドライバのインストール | 11 |
| 4 Apache Superset の利用方法 | 12 |
| 4.1 Apache Superset の画面 | 12 |
| 4.2 Apache Superset の利用フロー | 12 |
| 5 データベースの管理 | 14 |
| 5.1 データベースの追加 | 14 |
| 5.2 データセットの追加 | 17 |
| 6 チャートの管理 | 21 |
| 6.1 チャートの作成 | 21 |
| 7 ダッシュボードの管理 | 28 |
| 7.1 ダッシュボードの作成 | 28 |
| 7.2 ダッシュボードのデザイン変更 | 29 |
| 7.3 ダッシュボードの例 | 30 |
| 7.4 ダッシュボードでの絞り込み検索 | 31 |
| 7.5 ダッシュボードでのクエリ確認 | 33 |
| 7.6 ダッシュボードのその他機能 | 33 |
| 8 Apache Superset でのデータ操作 | 34 |
| 8.1 データのアップロード | 34 |
| 8.2 SQL Lab でのデータ操作 | 34 |
| 9 Apache Superset のユーザと権限管理 | 36 |
| 9.1 ユーザの管理 | 36 |
| 9.2 権限の管理 | 37 |
| 9.3 最小限のロールの作成検証 | 38 |
| 10 発展的な設定 | 41 |
| 10.1 非同期処理の実行設定 | 41 |
| 10.2 非同期処理関連機能の設定 | 41 |
| 10.3 非同期処理有効環境で利用可能な機能 | 46 |

| | |
|-------------------------------|-----------|
| 11 Apache Superset まとめ | 48 |
| 11.1 Apache Superset の特徴 | 48 |
| 11.2 Apache Superset の問題点 | 48 |
| 11.3 まとめ | 49 |

1 概要

1.1 本書について

本書は「Apache Superset」についての調査報告書です。

1.2 Apache Supersetについて

1.2.1 Apache Supersetとは

Apache Superset は、データの検索や可視化を行うための BI ツールの一種です。MySQL や PostgreSQL 等の RDBMS のほかに、約 40 種類のデータソースを扱うことができます。

1.2.2 ソフトウェア情報

Apache Superset の提供元・ライセンスは次の通りです。

- 提供元: Apache Foundation
- 公式サイト: <https://superset.apache.org/>
- ライセンス: Apache License 2.0
- 開発言語: python/javascript
- 調査時のバージョン: 1.4.0

1.3 機能概観

Apache Superset には、次のような機能があります。

- RDBMS と類似 DB・テーブルの登録
 - 30 種を超える DB の利用が可能
 - 検索対象としてテーブル毎の登録が可能
 - 複数のテーブルを結合 (Join) して、仮想的なテーブルとして登録が可能
 - * ただしデータベースを跨ぐことはできない
 - テーブルのカラムに表示名をつけることが可能
- DB のデータの可視化
 - テーブル・バーチャートなど約 60 種類の表現方法が存在
 - 単体のグラフ作成の他、ダッシュボードとしてまとめることが可能
- CSV 等でのデータアップロード・新規テーブル追加
 - 手元にあるデータをアップロードして、解析対象にすることが可能

- 権限の制御
 - ユーザに決め細かい権限の設定を行うことが可能

| 2 対応データベース一覧

Apache Superset は次のデータベース・クラウドサービスに対応しています。

- Amazon Athena
- Amazon Redshift
- Apache Drill
- Apache Druid
- Apache Hive
- Apache Impala
- Ascend.io
- Apache Kylin
- Apache Pinot
- Apache Solr
- Apache Spark SQL
- Clickhouse
- CockroachDB
- Dremio
- Elasticsearch
- Exasol
- Google BigQuery
- Google Sheets
- Hana
- IBM DB2
- IBM Netezza Performance Server
- MySQL
- Oracle
- Postgres
- Presto
- Snowflake
- Microsoft SQL Server
- Teradata
- Vertica
- Hologres
- Trino
- Rockset

- CrateDB
- Databricks
- Firebird
- Firebolt
- Kusto

I 3 Apache Superset のインストール方法

本章では Apache Superset のインストール方法について解説します。本内容は、次の公式マニュアルを参考に作成されています。

- <https://superset.apache.org/docs/installation/installing-superset-from-scratch>

II 3.1 環境

本書では次の環境を利用します。

- OS: Ubuntu 20.04 LTS
- KVM 仮想 + CPU4 コア + メモリ 4GB + ディスク: 10GB

II 3.2 依存関係ソフトウェアのインストール

Apache Superset を動作させるために必要になるソフトウェアやライブラリのインストールを行います。

```
$ sudo apt update
$ sudo apt install build-essential libssl-dev libffi-dev python3-dev python3-pip libsasl2-dev
  ↪ libldap2-dev redis
```

redis は動作上必須ではありませんが、Apache Superset を快適に動作させるためにキャッシング機能が必要になるためインストールします。なお、Ubuntu20.04LTS ではインストールと同時に起動するため、すぐ利用できます。

II 3.3 Python Virtual Environment の設定

III 3.3.1 superset ユーザの追加

Apache Superset を動作させるユーザの作成を行います。

```
$ sudo useradd -m -d /opt/superset/ -s /bin/bash superset
```

III 3.3.2 Python 環境の整備

Apache Superset を動作させるための Python 環境を整備します。これは Ubuntu20.04LTS の場合、ディストリビューション標準の Python モジュールと Apache Superset が要求する Python モジュールのバージョンに差があり、動作しないためです。

```
$ sudo apt install python3-virtualenv
$ sudo su - superset
$ virtualenv python-superset
$ . ./python-superset/bin/activate
```

上記で `/opt/superset/python-superset/` に Apache Superset 用の Python 環境が構築されます。

|| 3.4 MariaDB のインストール

Apache Superset の設定を保管する DB サーバを用意します。デフォルトでは SQLite が使われるため、単純に動作させるだけであれば、この設定は必須ではありません。

まず MariaDB をインストールします。また後の Python の mysqlclient モジュールで必要になるライブラリも合わせてインストールします。

```
$ sudo apt install mariadb-server libmariadbclient-dev
```

インストール後、MariaDB の初期化と Apache Superset 用のユーザとデータベースを作成します。

```
$ sudo mysql_secure_instration
$ mysql -u root -p
> CREATE DATABASE superset charset utf8mb4;
> CREATE USER 'superset'@'localhost' IDENTIFIED BY 'supersetPassword';
> GRANT ALL PRIVILEGES ON superset.* TO 'superset'@'localhost';
```

|| 3.5 Apache Superset のインストール

次のコマンドで、Apache Superset と依存関係のモジュールのインストールを行います。現状、依存モジュールのバージョン定義に問題があり、*Flask-WTF* のみバージョン指定のインストールが必要です。

```
$ pip install pip install Flask-WTF==0.14.3
$ pip install apache-superset pillow gevent mysqlclient
```

|| 3.6 Apache Superset の設定ファイルの作成

Apache Superset の設定ファイルは、任意の場所に配置することができます。本書では次のように作成します。

- `/opt/superset/etc/superset_config.py`

```
FLASK_APP = superset
ROW_LIMIT = 5000
SECRET_KEY = 'my_secret_key'
# SQLite を利用する場合の設定
# SQLALCHEMY_DATABASE_URI = 'sqlite:///opt/superset/.superset/superset.db'
SQLALCHEMY_DATABASE_URI = 'mysql://superset:supersetPassword@127.0.0.1/superset'

WTF_CSRF_ENABLED = True
WTF_CSRF_EXEMPT_LIST = []
WTF_CSRF_TIME_LIMIT = 60 * 60 * 24 * 365

# 多言語対応
LANGUAGES = {
```

```
"en": {"flag": "us", "name": "English"},  
"ja": {"flag": "jp", "name": "Japanese"},  
}  
  
CACHE_CONFIG = {  
    'CACHE_TYPE': 'redis',  
    'CACHE_DEFAULT_TIMEOUT': 60 * 60 * 24, # 1 day default (in secs)  
    'CACHE_KEY_PREFIX': 'superset_cache',  
    'CACHE_REDIS_PORT': 6379,  
    'CACHE_REDIS_DB': 1,  
    'CACHE_REDIS_URL': 'redis://127.0.0.1:6379/0',  
}  
  
DATA_CACHE_CONFIG = {  
    **CACHE_CONFIG,  
    "CACHE_KEY_PREFIX": "superset_data_cache",  
}
```

さらに初期設定を進めます。アプリケーションの動作に必要な環境変数を設定します。

```
$ export SUPERSET_CONFIG_PATH=/opt/superset/etc/superset_config.py
```

データベースの初期化を行います。

```
$ superset db upgrade
```

次に管理者ユーザを作成します。ユーザの情報の入力を求められるため、指示に従って入力します。

```
$ superset fab create-admin  
(中略)  
Username [admin]:  
User first name [admin]:  
User last name [user]:  
Email [admin@fab.org]: user@example.com  
Password: *****  
Repeat for confirmation: *****  
Recognized Database Authentications.  
Admin User admin created.
```

次にデフォルトのロールやパーミッションの初期化を行います。

```
$ superset init
```

次を実行すると、サンプルのデータを登録して、用意されたダッシュボードやグラフを確認することができます。必須ではありません。

```
$ superset load-examples
```

最後に Superset を起動します。以下は Apache Superset の公式マニュアルで推奨されている gunicorn を利用した動作方法です。ただし、worker の数 (-w) などは、利用環境によって調整が必要です。

gunicorn は Apache Superset のインストール時に自動的にインストールされるため、すぐ利用可能です。

```
$ gunicorn -w 10 -k gevent --timeout 120 -b 0.0.0.0:8088 --limit-request-line 0 --limit-request-field_size 0 "superset.app:create_app()"
```

なお、systemd のユニットファイルは次のようになります。

- /etc/systemd/system/superset.service

```
[Unit]
Description=Superset
After=network.target

[Install]
WantedBy=multi-user.target

[Service]
Type=simple
User=superset
Group=superset
WorkingDirectory=/opt/superset
Environment=SUPERSET_CONFIG_PATH=/opt/superset/etc/superset_config.py
Environment=VIRTUAL_ENV=/opt/superset/python-superset
Environment=PATH=/opt/superset/python-superset/bin
ExecStart=/opt/superset/python-superset/bin/gunicorn -w 10 -k gevent --timeout 120 -b 0.0.0.0:8088 --limit-request-line 0 --limit-request-field_size 0 "superset.app:create_app()"
LimitNOFILE=32768
```

上記設定を行うと、次のような URL で Apache Superset を利用することができます。

- <http://IPAddr:8088/>



|| 3.7 データベースドライバのインストール

Apache Superset は多くのデータベースに対応しています。扱えるデータベースは、基本的に SQL 互換のデータベースです。

対応可能なデータベースは多いのですが、デフォルトの状態では SQLite しか扱うことができません。様々なデータベースを扱うためには、ドライバのインストールが必要です。

例えば、MySQL であれば、次のように mysqlclient ライブラリをインストールする必要があります。

```
$ pip install mysqlclient
```

その他、いくつかのドライバのインストール方法を例示します。

```
# PostgreSQL
$ pip install psycopg2

# Google Big Query
$ pip install pybigquery

# Elasticsearch
$ pip install elasticsearch-dbapi
```

どのデータベースのドライバも基本的に pip コマンドを利用してインストール可能です。データベース毎のドライバのインストール方法の詳細は次の URL に記載があります

- <https://superset.apache.org/docs/databases/installing-database-drivers>

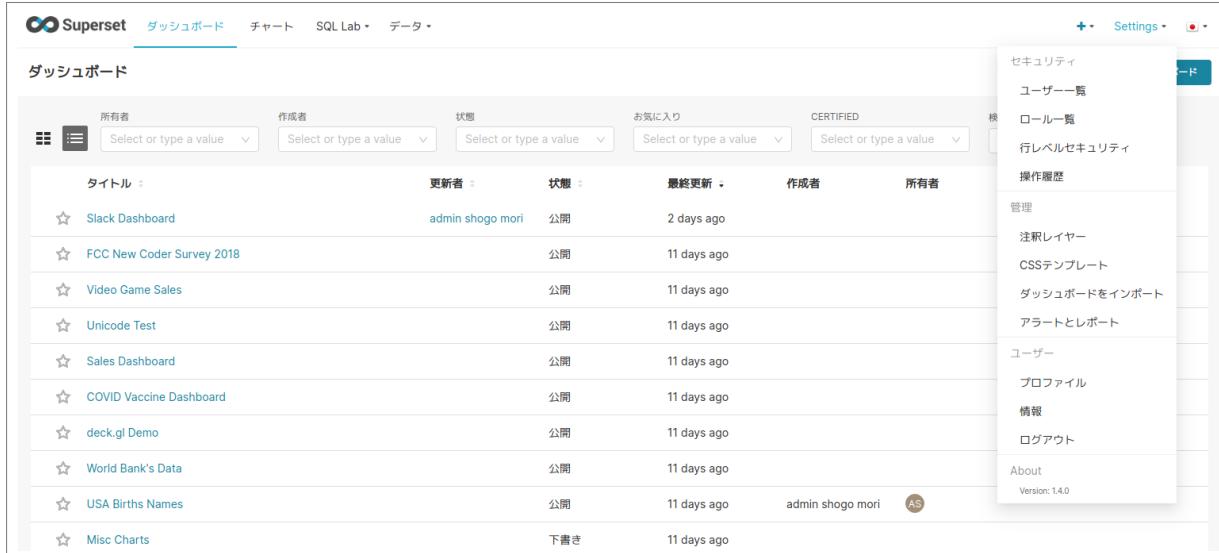
なお Elasticsearch は、SQL 互換ではありません。しかし elasticsearch-dbapi というドライバが、SQL 互換のクエリを使って Elasticsearch にリクエストを送信できるため、Apache Superset でも取扱いが可能になっています。

4 Apache Superset の利用方法

本章では、Apache Superset の基本的な利用方法について解説します。なお、解説のため Apache Superset 付属のサンプルのデータを利用します。

4.1 Apache Superset の画面

Apache Superset のログイン後の画面は次のようにになっています。基本的には上部のメニューを選択して、各画面に移動します。また画面右上の Settings から、ユーザ自身や管理者向けの設定を行うことができます。



The screenshot shows the Apache Superset dashboard. At the top, there are navigation links: 'Superset' (with a logo), 'ダッシュボード' (selected), 'チャート', 'SQL Lab', and 'データ'. On the right, there is a 'Settings' dropdown menu with various options like 'セキュリティ', 'ユーザー', '操作履歴', etc. The main area is titled 'ダッシュボード' and lists ten sample dashboards. Each entry includes a star icon, the dashboard name, the owner ('所有者'), creator ('作成者'), status ('状態'), last updated ('最終更新'), author ('作成者'), and owner ('所有者').

| タイトル | 更新者 | 状態 | 最終更新 | 作成者 | 所有者 |
|---------------------------|------------------|-----|-------------|------------------|-----|
| Slack Dashboard | admin shogo mori | 公開 | 2 days ago | | |
| FCC New Coder Survey 2018 | | 公開 | 11 days ago | | |
| Video Game Sales | | 公開 | 11 days ago | | |
| Unicode Test | | 公開 | 11 days ago | | |
| Sales Dashboard | | 公開 | 11 days ago | | |
| COVID Vaccine Dashboard | | 公開 | 11 days ago | | |
| deck.gl Demo | | 公開 | 11 days ago | | |
| World Bank's Data | | 公開 | 11 days ago | | |
| USA Births Names | | 公開 | 11 days ago | admin shogo mori | |
| Misc Charts | | 下書き | 11 days ago | | |

各メニューの機能は次の通りです。

- ダッシュボード: ダッシュボードの管理
- チャート: 単体のグラフ等の管理
- SQL Lab: SQL クエリの実行・登録
- データ: データベース・データセットの管理
- Setting: ユーザ・ロール等の管理

4.2 Apache Superset の利用フロー

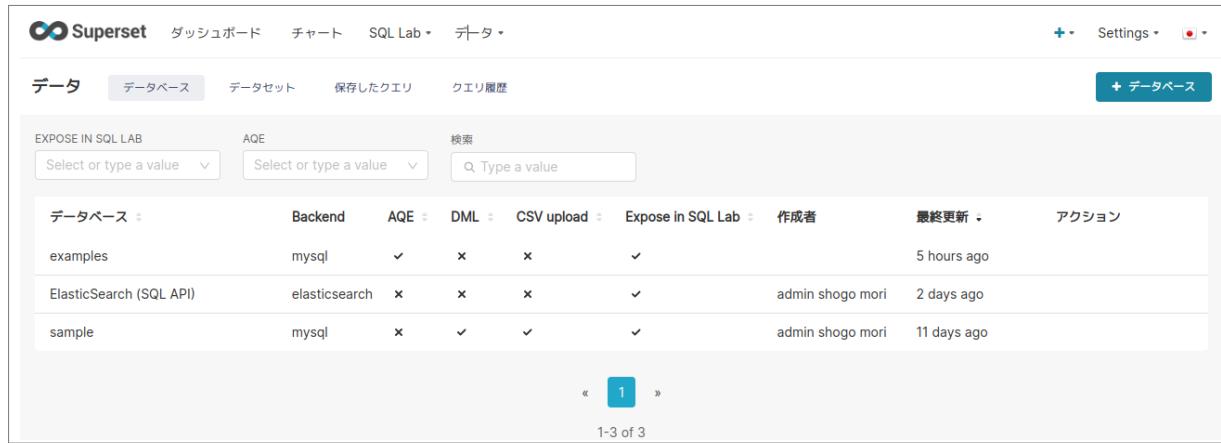
Apache Superset は、次のように段階を追って利用します。

1. データベースの登録
2. データセットの登録
3. SQL Lab でのデータ検索
4. チャートでの可視化
5. ダッシュボードの作成

この次の章から順を追って、解説を進めていきます。

5 データベースの管理

Apache Superset を使ってデータの解析を行うためには、まずデータベースの登録を行います。次はデータベースの管理画面です。



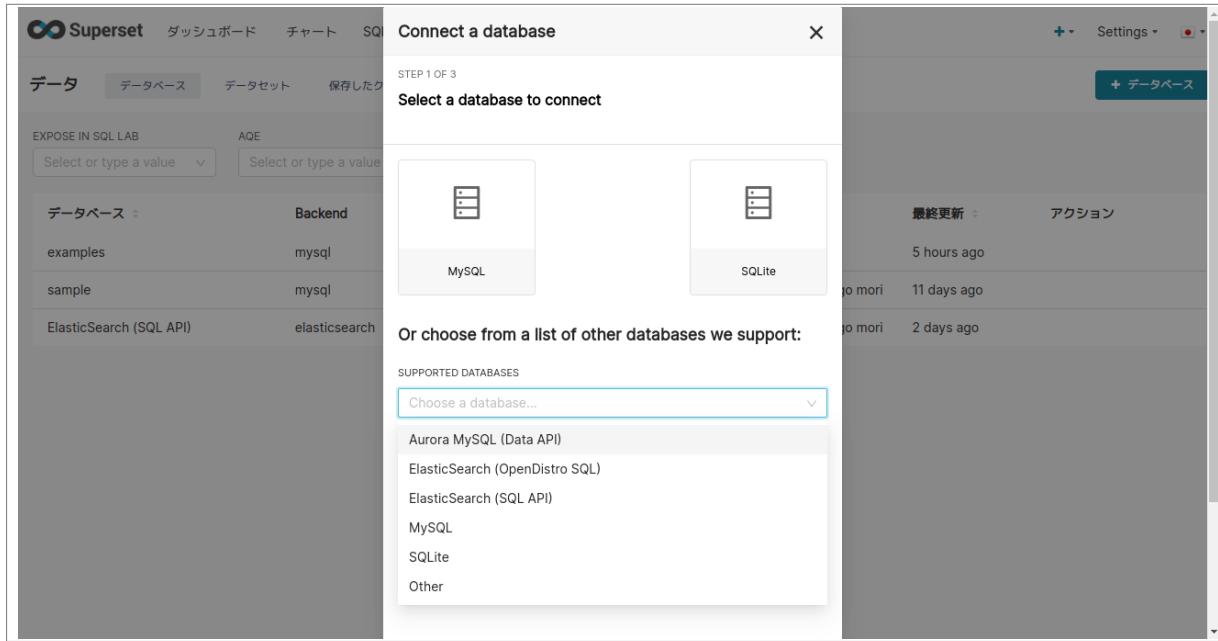
| データベース | Backend | AQE | DML | CSV upload | Expose in SQL Lab | 作成者 | 最終更新 | アクション |
|-------------------------|---------------|-----|-----|------------|-------------------|------------------|-------------|-------|
| examples | mysql | ✓ | ✗ | ✗ | ✓ | admin shogo mori | 5 hours ago | |
| ElasticSearch (SQL API) | elasticsearch | ✗ | ✗ | ✗ | ✓ | admin shogo mori | 2 days ago | |
| sample | mysql | ✗ | ✓ | ✓ | ✓ | admin shogo mori | 11 days ago | |

上記画像では、登録済みのデータベースが表示されています。各項目の意味は次の通りです。

- データベース: データベースの名前
- Backend: データベースの種類
- AQE: 非同期クエリの有効/無効
- DML: データ操作クエリ (INSERT や DELETE) の有効/無効
- CSV upload: CSV でのデータ登録の有効/無効
- Expose in SQL Lab: SQL Lab での操作の有効/無効

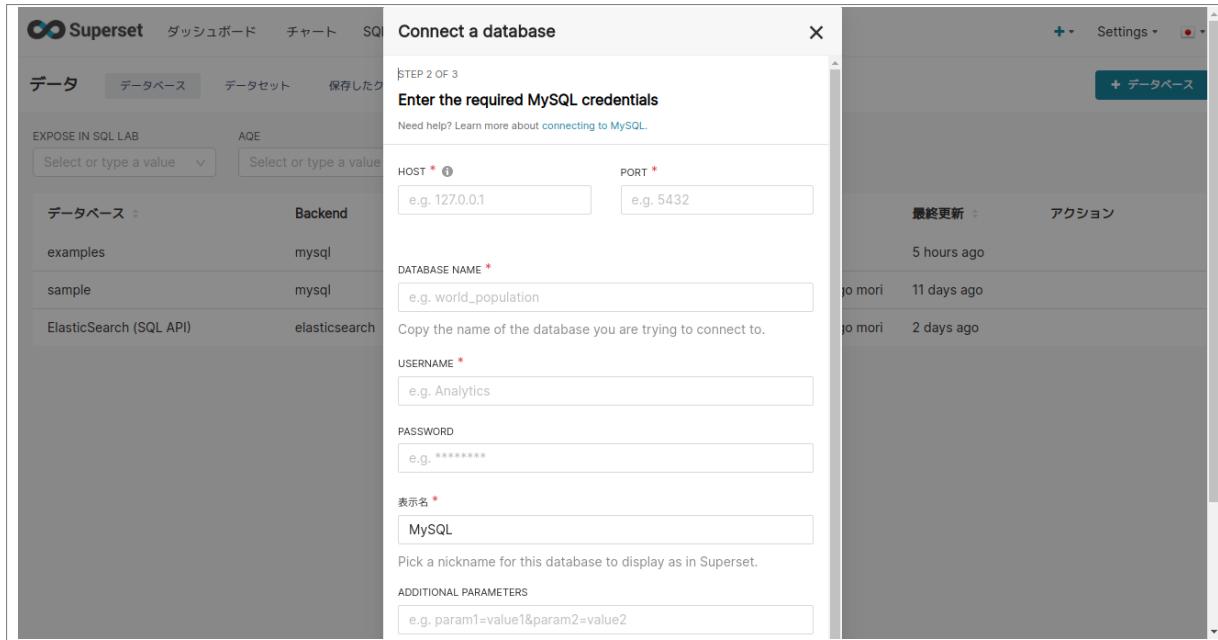
5.1 データベースの追加

右上の [+データベース] から新たなデータベースを追加することができます。データベースの追加画面は、次の通りです。

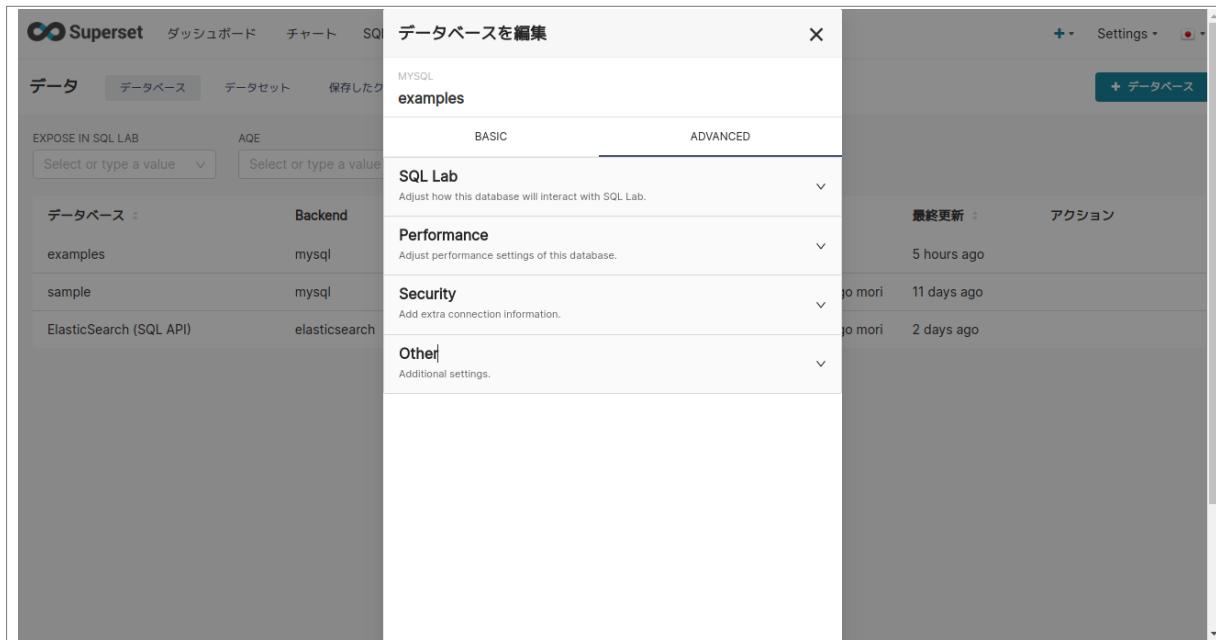


まずデータベースの種類を選択します。Apache Superset のサーバにドライバがインストールされていれば、自動的に対応データベースが表示されます。

種類を選択すると、接続情報の入力画面が表示されます。以下は MySQL を選択した場合の画面です。



入力処理を進めると、発展的な設定も入力が可能です。



ここでは、次のような項目の設定を行うことができます。

- SQL Lab: SQL Lab での操作許可設定
 - テーブルの作成の許可
 - VIEW の作成の許可
 - INSERT や DELETE, UPDATE などの更新系クエリの許可
 - 全てのテーブルのメタデータの取得の許可
 - PostgreSQL や Presto のクエリコスト計算機能の有効化
 - SQL Lab のクエリの可視化の有効化
- パフォーマンス設定
 - 各種キャッシュのタイムアウト時間の設定
 - 非同期でのクエリ実行の有効化
- セキュリティ関連設定
 - クラウドサービスデータベース (Big Query など) の認証情報の登録
 - CSV アップロードの許可
- その他
 - 任意のメタデータの設定
 - バージョンの設定

|| 5.2 データセットの追加

データベースの登録後、データセットを追加します。データセットは、RDBMS のテーブルを抽象化した概念です。

データセットとして追加する意義は次の通りです。

- テーブルのカラム毎に表示名などのメタデータを登録可能
- 可視化のために利用する関数 (COUNT など) を、予めわかりやすい名前で登録可能
- 存在するテーブルだけでなく、仮想的なテーブルを登録可能
 - 複数のテーブルを JOIN して扱うなど

つまりデータセットは、データベースのテーブルを、そのまま扱うよりもよりユーザビリティを高めることを目的としたものです。

以下はデータセットの一覧画面です。

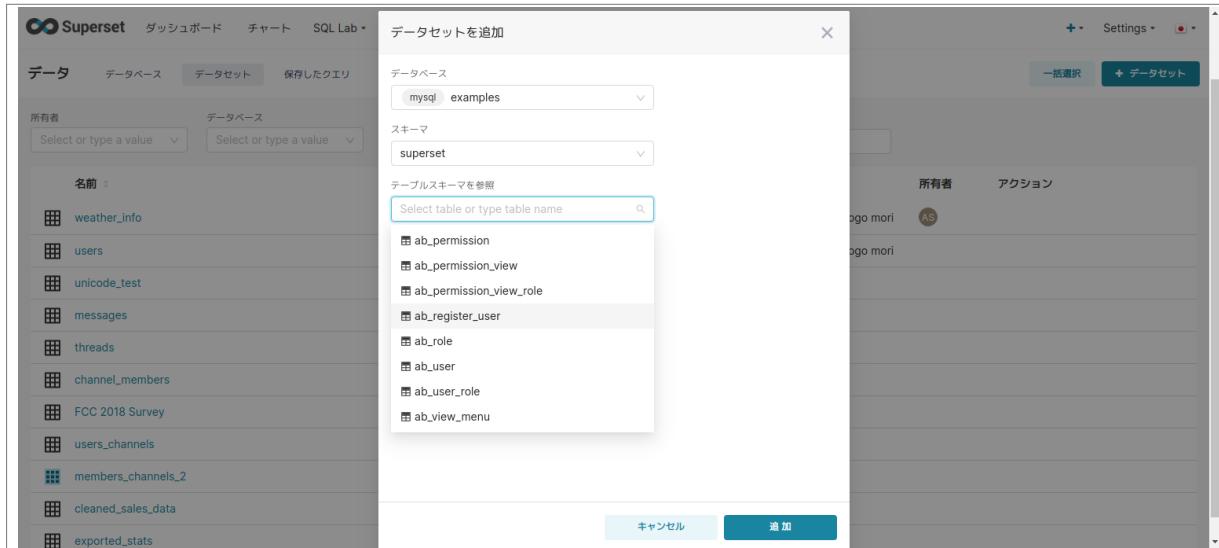
The screenshot shows the Superset Data Set list interface. At the top, there are tabs for 'データ' (Data), 'データベース' (Database), 'データセット' (Dataset), '保存したクエリ' (Saved Query), and 'クエリ履歴' (Query History). The 'データセット' tab is selected. Below the tabs are search and filter fields for '所有者' (Owner), 'データベース' (Database), 'スキーマ' (Schema), 'タイプ' (Type), and a search bar. A large table lists the datasets:

| 名前 | タイプ | データベース | スキーマ | 最終更新 | 更新者 | 所有者 | アクション |
|-----------------|----------|----------|----------|-------------|------------------|-----|-----------------------|
| weather_info | Physical | sample | sample | 11 days ago | admin shogo mori | AS | <button>edit</button> |
| users | Physical | examples | superset | 11 days ago | admin shogo mori | | <button>edit</button> |
| unicode_test | Physical | examples | superset | 11 days ago | | | <button>edit</button> |
| messages | Physical | examples | superset | 11 days ago | | | <button>edit</button> |
| threads | Physical | examples | superset | 11 days ago | | | <button>edit</button> |
| channel_members | Physical | examples | superset | 11 days ago | | | <button>edit</button> |

ポイントとなる項目は次です。

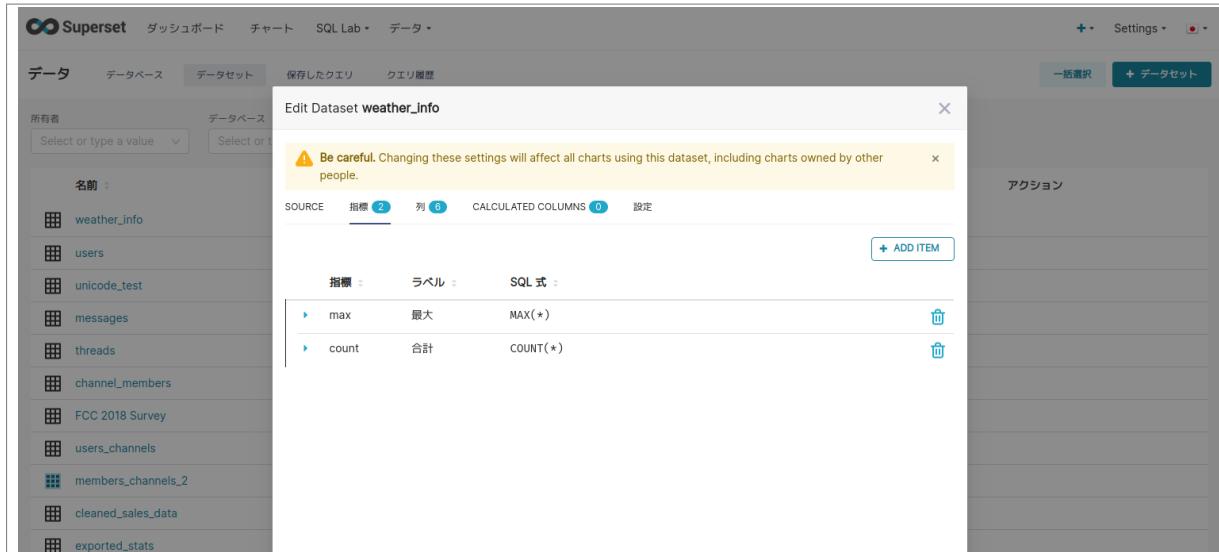
- 名前: データセットの名前
- タイプ
 - Physical はデータベースに存在するテーブルです。
 - Virtual は仮想的に作成されたテーブルです
- データベース
 - データベースです
- スキーマ
 - データセットの元になるテーブルです

次はデータセットの追加画面です。

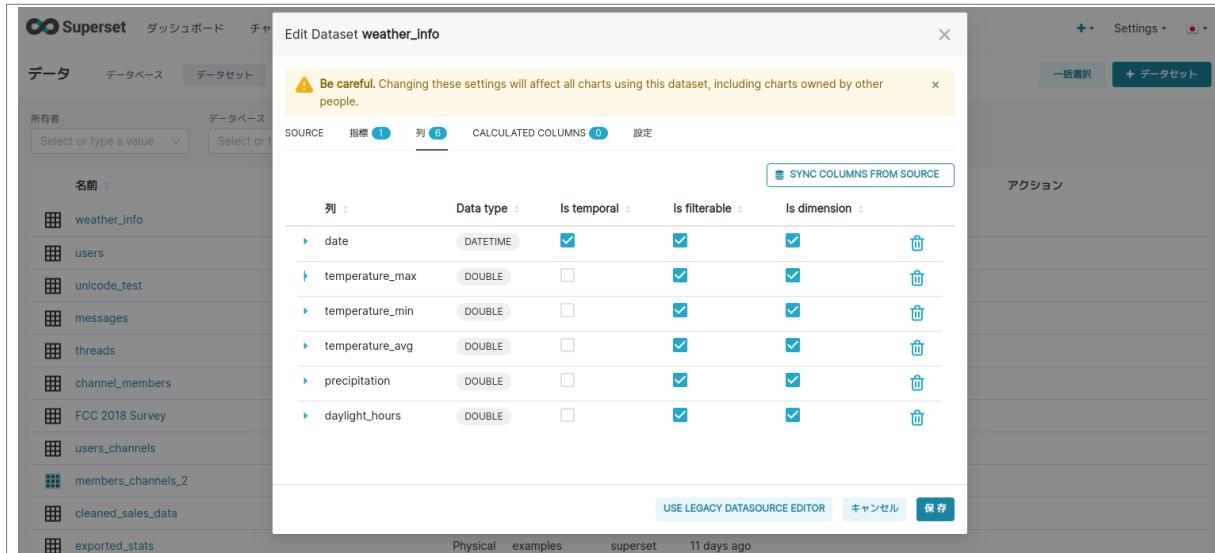


まず追加するデータベースやテーブルを選択します。追加すると、さらに詳細な項目の設定を行うことができます。

まず指標には、可視化で利用する関数と表示ラベルを登録できますなお指標は原文では Metrics です。



次に列には、テーブルのカラムの動作や表示名などを登録することができます。



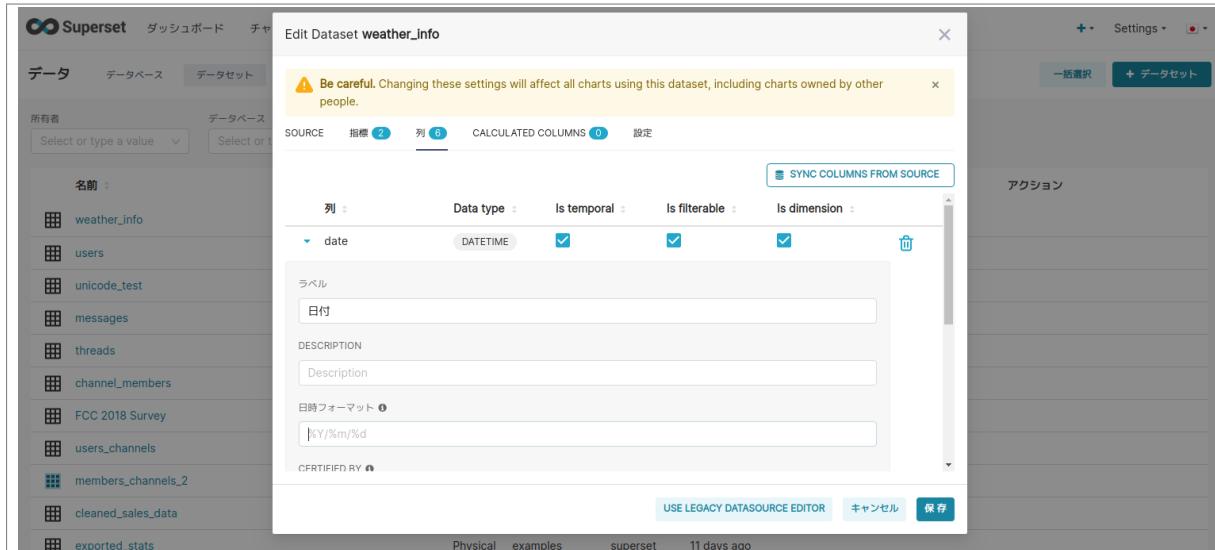
The screenshot shows the 'Edit Dataset weather_info' dialog in Superset. The dataset has 6 columns: date, temperature_max, temperature_min, temperature_avg, precipitation, and daylight_hours. The 'date' column is marked as the primary key (PK). All columns are marked as temporal, filterable, and dimension.

| 列 | Data type | Is temporal | Is filterable | Is dimension |
|-----------------|-----------|-------------------------------------|-------------------------------------|-------------------------------------|
| date | DATETIME | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| temperature_max | DOUBLE | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| temperature_min | DOUBLE | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| temperature_avg | DOUBLE | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| precipitation | DOUBLE | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| daylight_hours | DOUBLE | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

カラムの動作設定は次の通りです。

- Data type: データの型
- is temporal: 時間軸として利用できるかどうか
- is filterable: データのフィルタリングが可能化どうか
- is dimension: X/Y 軸の値として利用可能かどうか

またカラム毎に、表示名や説明、日付フォーマットを追加することもできます。



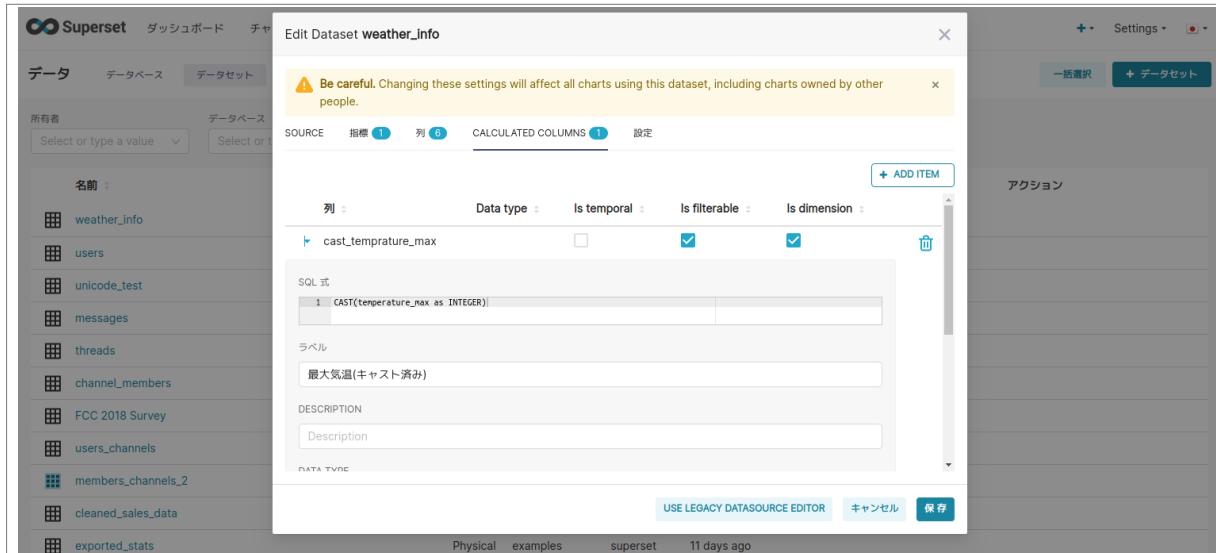
The screenshot shows the 'Edit Dataset weather_info' dialog in Superset. The 'date' column has been configured with a label ('日付'), a description ('Description'), and a date format ('日時フォーマット') set to 'YYYY/MM/DD'.

| 列 | Data type | Is temporal | Is filterable | Is dimension |
|------|-----------|-------------------------------------|-------------------------------------|-------------------------------------|
| date | DATETIME | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

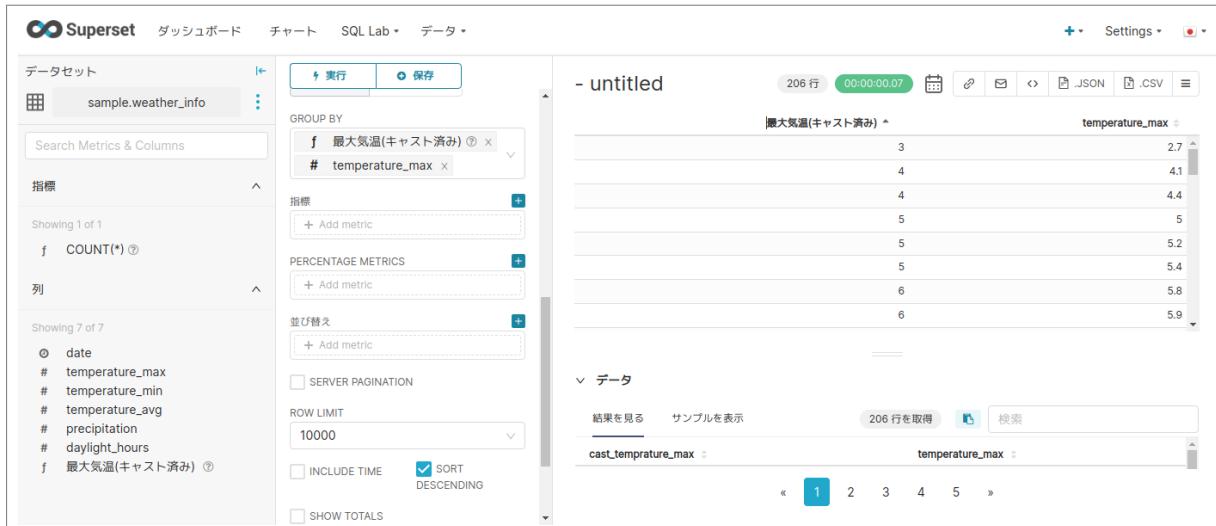
Additional configuration for the 'date' column:

- ラベル: 日付
- DESCRIPTION: Description
- 日時フォーマット: YYYY/MM/DD

CALCULATED COLUMNS は、事前に SQL で加工・計算したデータを仮想的なカラムとして登録する機能です。



上記は、最高気温を表す `temperatur_max` の小数の値を、整数に変換して登録している例です。元のカラムと、新しく作ったカラムを可視化すると、次のように変化しているのがわかります。



設定の項目には、データセットの情報や動作に関わる設定を行うことができます。

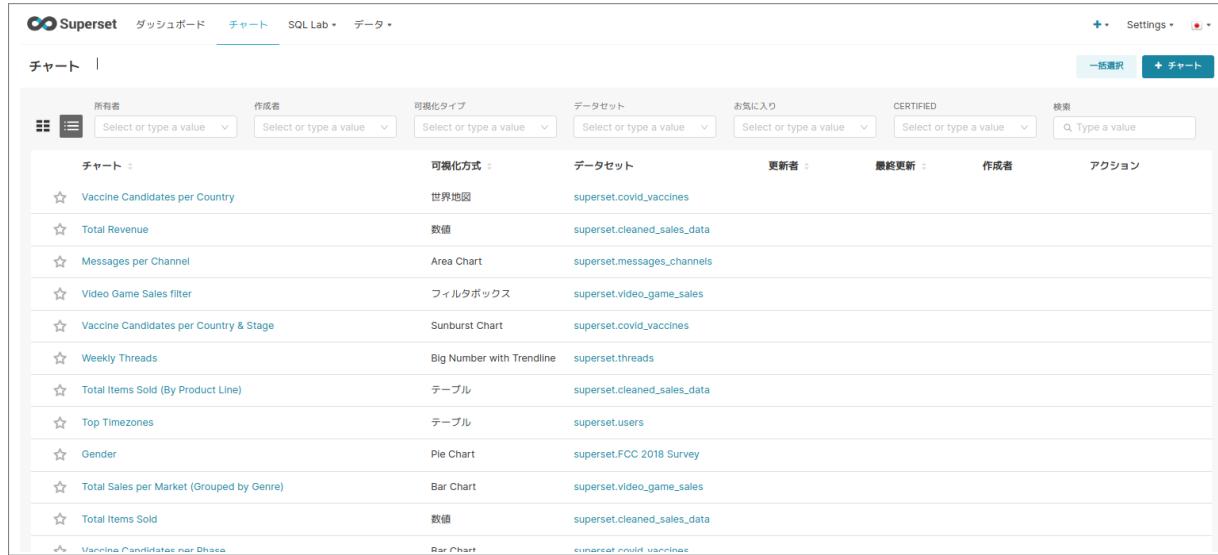
- 説明の追加
- 他のデータセットへのリダイレクト設定 (データセットが古くなった場合など)
- オートコンプリートフィルタの有効化
 - カラムの値 (国名のリスト等) を DB から取得して、検索条件に表示する設定

データセットを追加すると、データの可視化を行えるようになります。

6 チャートの管理

データセットを登録後、データセットから可視化を行うことができます。この可視化の機能を Apache Superset ではチャートと呼びます。

以下がチャートの管理画面です。



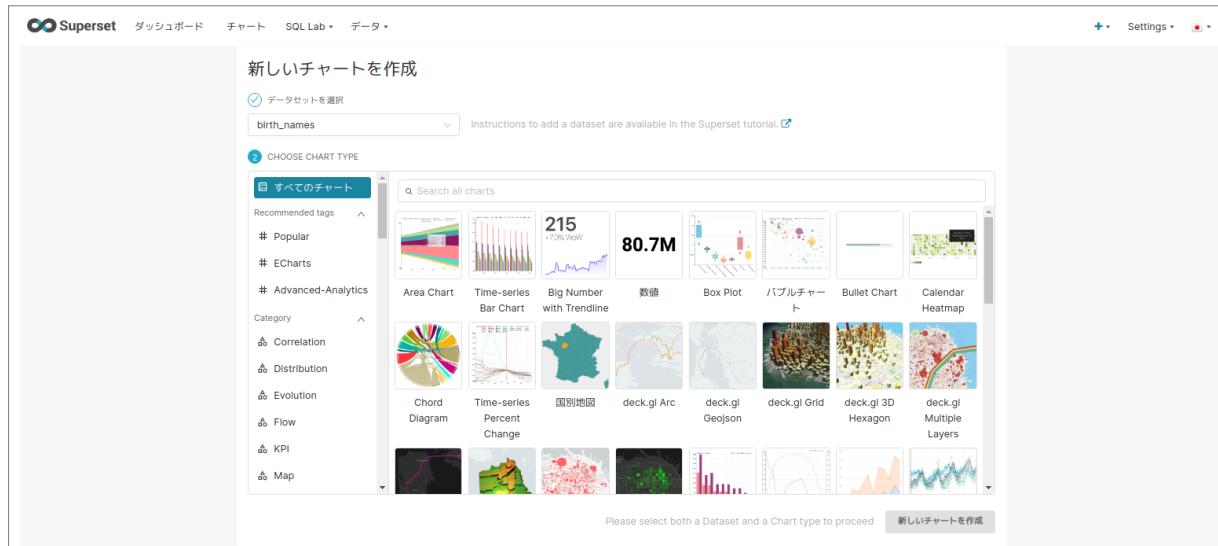
The screenshot shows the Apache Superset interface under the 'チャート' tab. It displays a list of registered charts with the following details:

| チャート | 可視化方式 | データセット | 更新者 | 最終更新 | 作成者 | アクション |
|---|---------------------------|-----------------------------|-----|------|-----|---------|
| Vaccine Candidates per Country | 世界地図 | superset.covid_vaccines | | | | [+チャート] |
| Total Revenue | 数値 | superset.cleaned_sales_data | | | | [+チャート] |
| Messages per Channel | Area Chart | superset.messages_channels | | | | [+チャート] |
| Video Game Sales filter | フィルタボックス | superset.video_game_sales | | | | [+チャート] |
| Vaccine Candidates per Country & Stage | Sunburst Chart | superset.covid_vaccines | | | | [+チャート] |
| Weekly Threads | Big Number with Trendline | superset.threads | | | | [+チャート] |
| Total Items Sold (By Product Line) | テーブル | superset.cleaned_sales_data | | | | [+チャート] |
| Top Timezones | テーブル | superset.users | | | | [+チャート] |
| Gender | Pie Chart | superset.FCC 2018 Survey | | | | [+チャート] |
| Total Sales per Market (Grouped by Genre) | Bar Chart | superset.video_game_sales | | | | [+チャート] |
| Total Items Sold | 数値 | superset.cleaned_sales_data | | | | [+チャート] |
| Vaccine Candidates per Phase | Bar Chart | superset.covid_vaccines | | | | [+チャート] |

右上の [+ チャート] からチャートの追加を行うことができます。

6.1 チャートの作成

チャートの作成画面に移動すると、データセットと可視化のタイプを選択する画面が表示されます。



The screenshot shows the Apache Superset interface under the 'チャート' tab, specifically the '新しいチャートを作成' (Create New Chart) section. It displays the following interface elements:

- A checkbox labeled 'データセットを選択' (Select Dataset) with 'birth_names' selected.
- A search bar labeled 'Search all charts'.
- A sidebar titled 'すべてのチャート' (All Charts) containing categories like 'Popular', 'ECharts', 'Advanced-Analytics', etc., with 'Correlation' currently expanded.
- A grid of chart preview cards, including Area Chart, Time-series Bar Chart, Big Number with Trendline, Box Plot, Bullet Chart, Calendar Heatmap, Chord Diagram, Time-series Percent Change, 国別地図, deck.gl Arc, deck.gl Geojson, deck.gl Grid, deck.gl 3D Hexagon, and deck.gl Multiple Layers.
- A message at the bottom: 'Please select both a Dataset and a Chart type to proceed'.
- A button at the bottom right: '新しいチャートを作成' (Create New Chart).

約 60 種類のタイプの中から選択することができます。ただタイプは後から変更することも可能なので、まずはデータの内容がわかりやすいテーブル(表)から可視化を開始します。

なお選択したデータセットは、Apache Superset のサンプルとして登録されている birth_names というテーブルです。テーブルの内容は次のように、年代毎の名前の統計情報になっています。

```
MariaDB [superset]> select * from birth_names limit 10;
+-----+-----+-----+-----+-----+
| ds   | gender | name  | num   | state | num_boys | num_girls |
+-----+-----+-----+-----+-----+
| 1965-01-01 00:00:00 | boy    | Aaron | 369  | CA    | 369     | 0      |
| 1965-01-01 00:00:00 | girl   | Amy   | 494  | CA    | 0       | 494    |
| 1965-01-01 00:00:00 | girl   | Andrea | 607  | CA    | 0       | 607    |
| 1965-01-01 00:00:00 | boy    | Andrew | 933  | CA    | 933     | 0      |
| 1965-01-01 00:00:00 | girl   | Angela | 1066 | CA    | 0       | 1066   |
| 1965-01-01 00:00:00 | girl   | Anna   | 564  | CA    | 0       | 564    |
| 1965-01-01 00:00:00 | boy    | Anthony | 1928 | CA    | 1928    | 0      |
| 1965-01-01 00:00:00 | girl   | Barbara | 922  | CA    | 0       | 922    |
| 1965-01-01 00:00:00 | boy    | Bradley | 395  | CA    | 395     | 0      |
| 1965-01-01 00:00:00 | girl   | Brenda | 961  | CA    | 0       | 961    |
+-----+-----+-----+-----+-----+
```

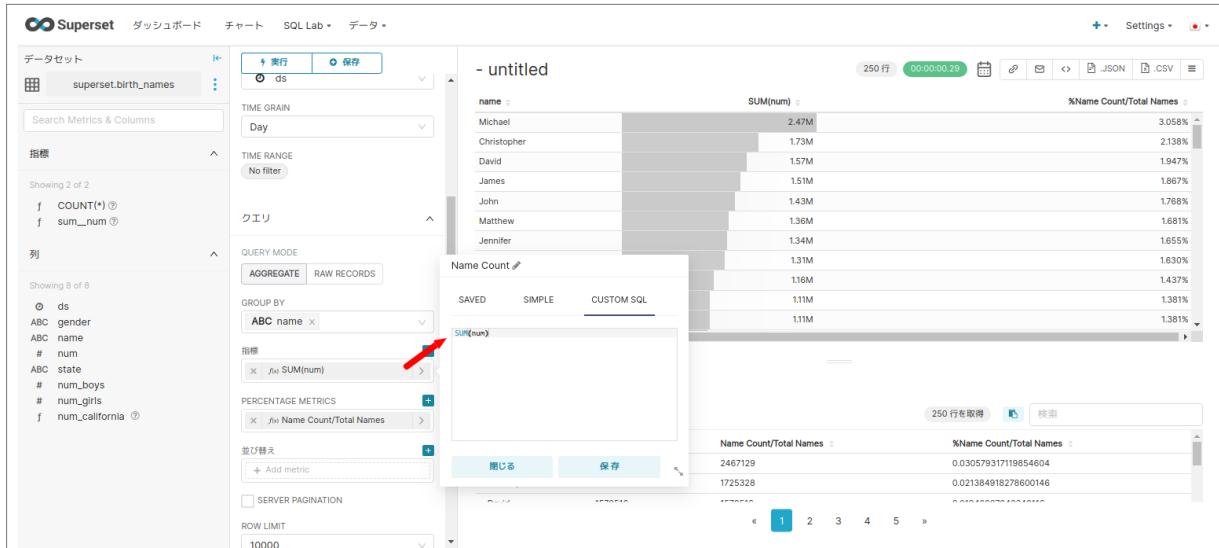
この情報で様々なチャートを作ることができます。

6.1.1 集計テーブルの作成

テーブルの場合、2つのモードを選ぶことができます。

- AGGREGATE: 集計テーブル
- RAW RECORDS: データセットの情報をそのまま表示するテーブル

以下は AGGREGATE を指定している例です。

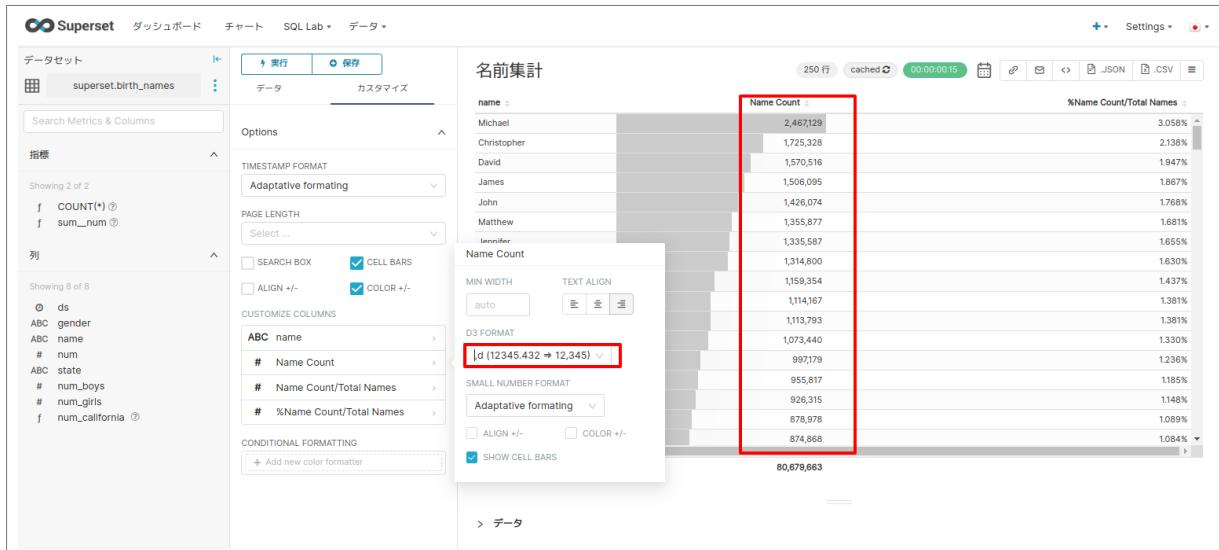


集計テーブルの場合、集計対象（グルーピングする）のカラムを [GROUP BY] の項目に設定します。

また、[指標] に集計のためのクエリを入力します。上記の例では全ての年代の名前の数の総計を計算するために、SUM 関数を利用しています。

さらに [PERCENTAGE METRICS] では、データ全体の割合を算出するための設定が行えます。ここで設定するのは全体の母数にあたる値の算出方法です。単純に SUM(num) という関数が登録されており、全体の num の値を計算しています。

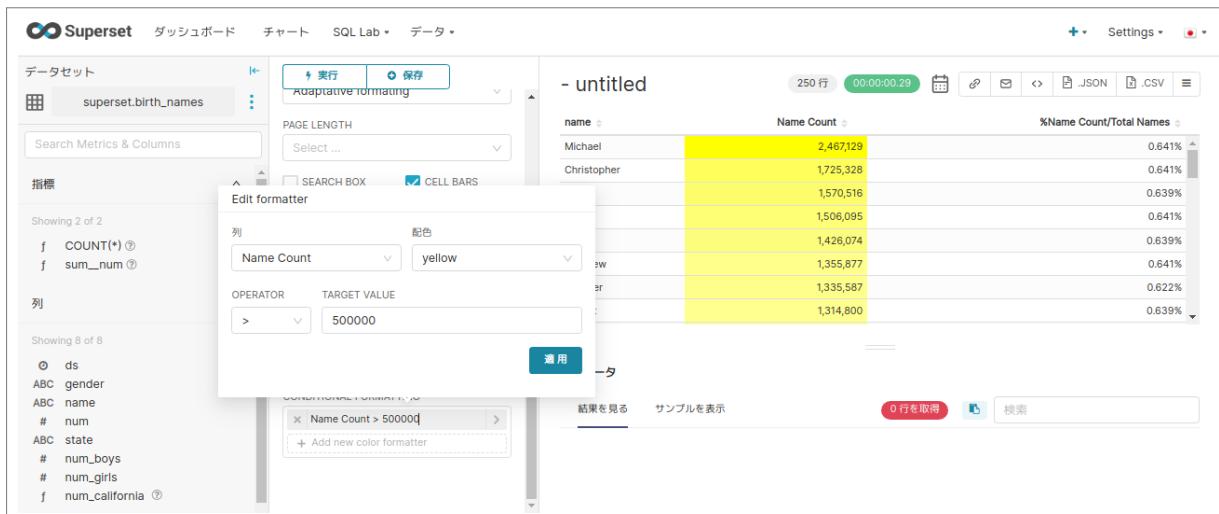
さらにカスタマイズのタブで、カラム毎の細かい表示設定を行うこともできます。例えば、現在は1000件を超える値が1Kや1Mなどのまとめられた単位になっています。この表示を変更することも可能です。



The screenshot shows the Superset interface with a data table titled "名前集計". A context menu is open over the first column, specifically the "Name Count" column. The menu is titled "D3 FORMAT" and contains a dropdown for "Number Format" set to "10 (12345.432 → 12,345)". The entire column header row is highlighted with a red box.

| name | Name Count | %Name Count/Total Names |
|-------------|------------|-------------------------|
| Michael | 2,467,129 | 3.05% |
| Christopher | 1,725,328 | 2.13% |
| David | 1,570,516 | 1.94% |
| James | 1,506,095 | 1.86% |
| John | 1,426,074 | 1.76% |
| Matthew | 1,355,877 | 1.68% |
| Jennifer | 1,335,587 | 1.65% |
| Sarah | 1,314,800 | 1.63% |
| Alice | 1,159,354 | 1.43% |
| Robert | 1,114,167 | 1.38% |
| Emily | 1,113,793 | 1.381% |
| Daniel | 1,073,440 | 1.33% |
| Olivia | 997,179 | 1.236% |
| William | 955,817 | 1.185% |
| Isabella | 926,315 | 1.148% |
| Benjamin | 878,978 | 1.089% |
| Madison | 874,868 | 1.084% |
| | 80,679,663 | |

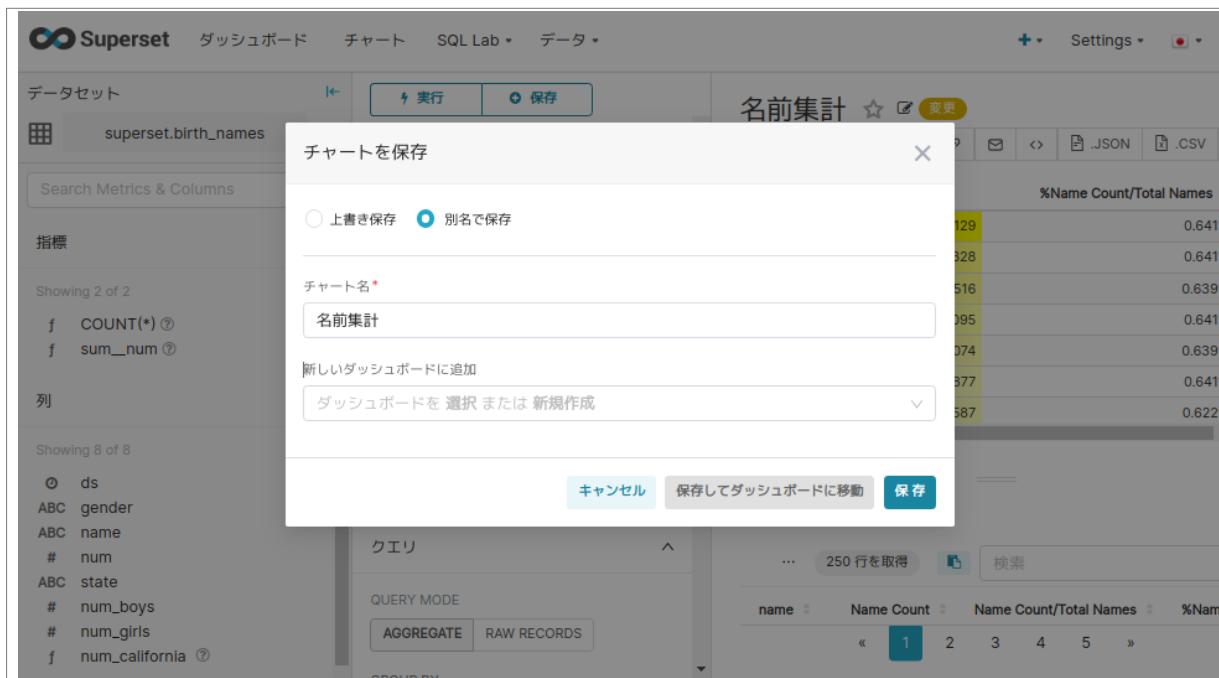
また基準値を超える値などの条件で、列の背景をグラデーションにすることもできます。



The screenshot shows the Superset interface with a data table titled "- untitled". A context menu is open over the first column, specifically the "Name Count" column. The menu is titled "Edit formatter" and shows a "TARGET VALUE" of "500000". The background of the entire column header row is yellow, indicating it has been formatted according to the rule "Name Count > 500000".

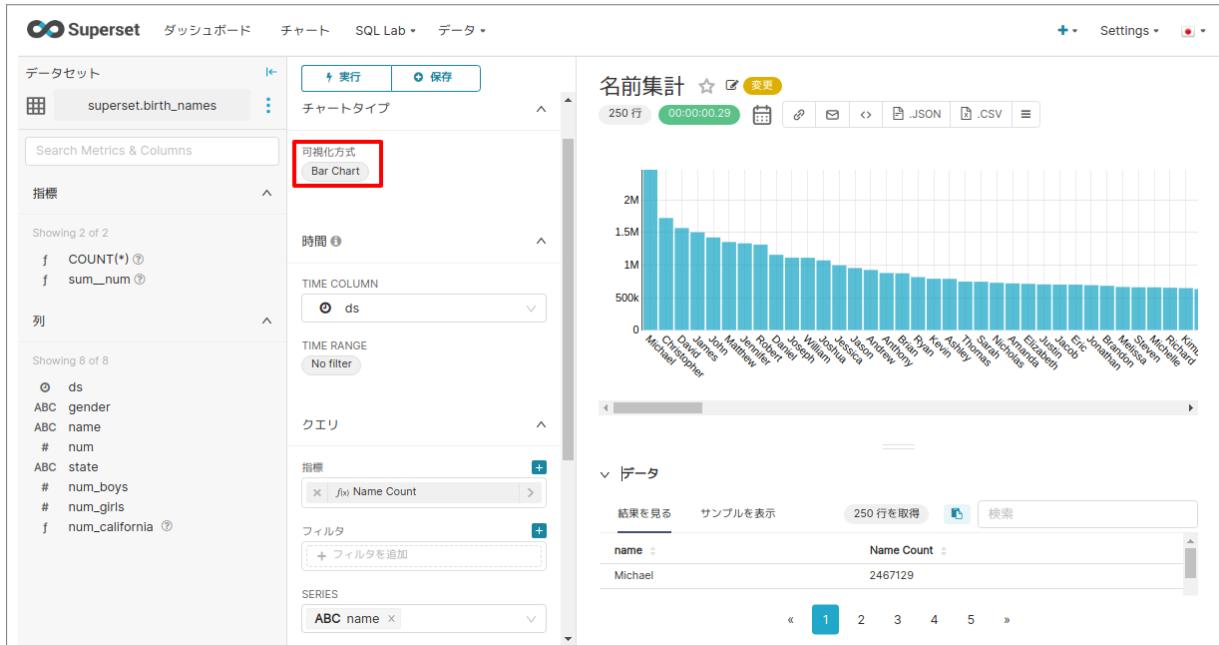
| name | Name Count | %Name Count/Total Names |
|-------------|------------|-------------------------|
| Michael | 2,467,129 | 0.641% |
| Christopher | 1,725,328 | 0.641% |
| David | 1,570,516 | 0.639% |
| James | 1,506,095 | 0.641% |
| John | 1,426,074 | 0.639% |
| Matthew | 1,355,877 | 0.622% |
| Jennifer | 1,335,587 | 0.622% |
| Sarah | 1,314,800 | 0.639% |

可視化した表やグラフは、画面上部の実行ボタンで最作成が可能です。また保存ボタンを押すと、保存して後から見返したり、ダッシュボードに登録することができるようになります。



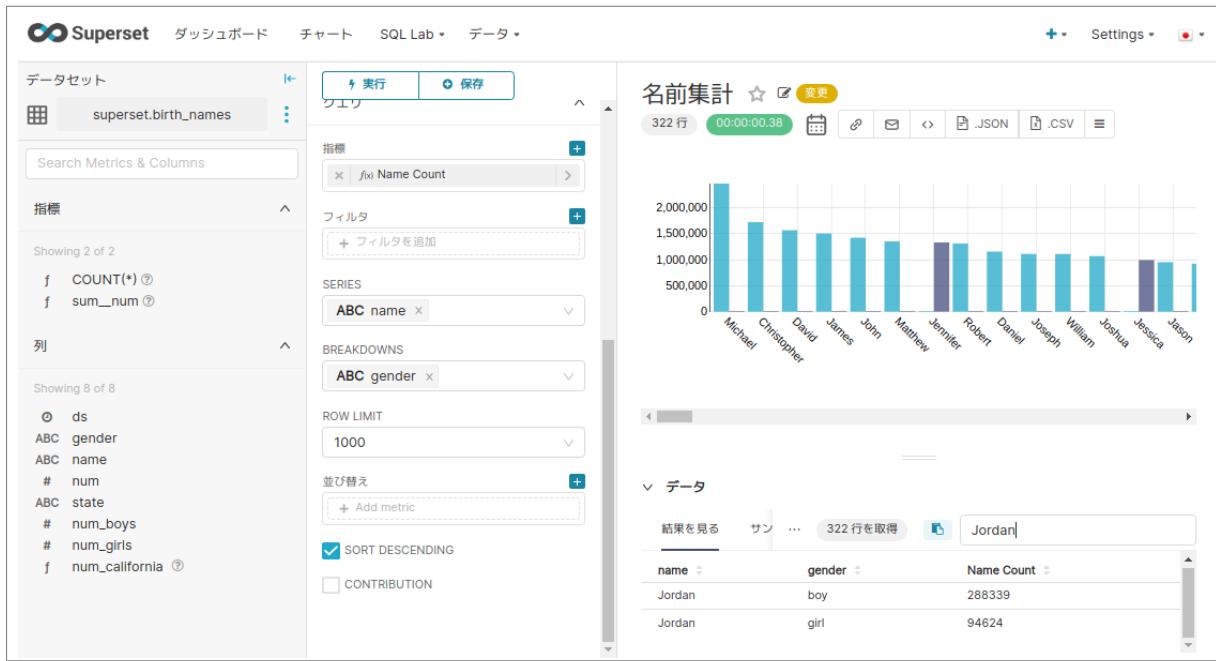
6.1.2 Bar Chart(棒グラフ)の作成

上記の名前集計のチャートの、可視化タイプを Bar Chart に変更した場合、次のようにになります。



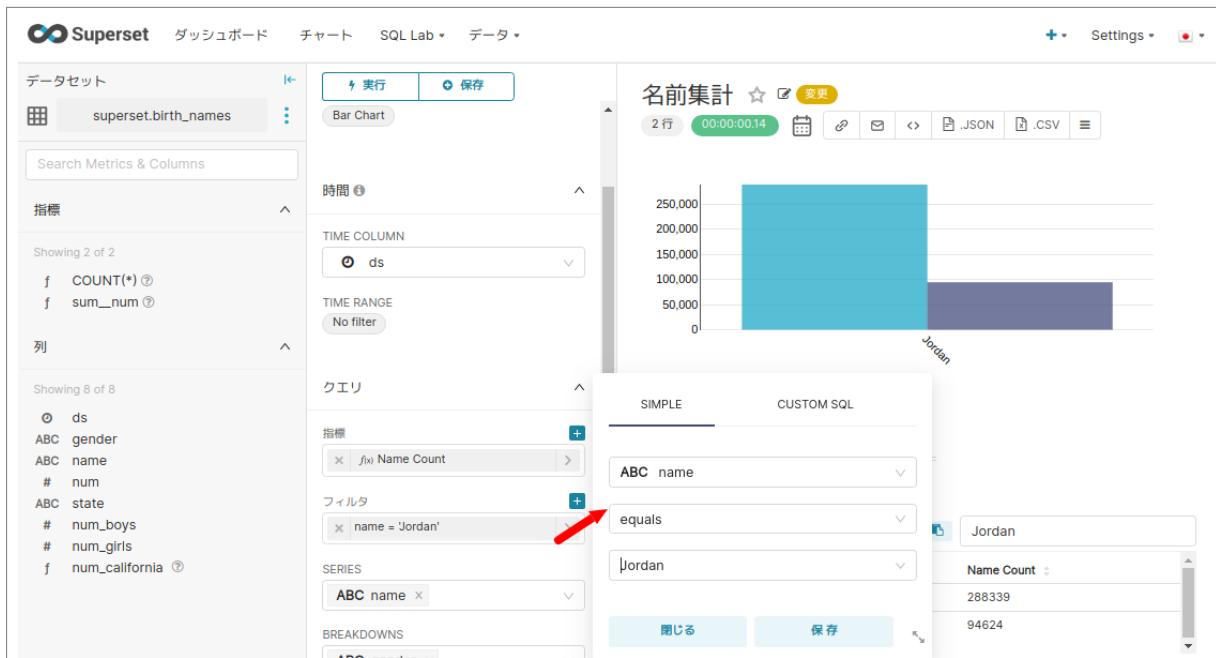
自動的に表示が切り替わり、Bar Chart 専用のオプションが現れます。Bar Chart では次のような操作が可能です。

まず BREAKDOWNS です。これは統計情報を細分化するためのオプションです。例えば名前の集計に性別の要素を加えて分類した場合、次のようにになります。



青色が男性、紫が女性というようにデータが細分化されてグラフ化されています。

さらにフィルタの機能を使って、データを絞り込むこもできます。以下は Jordan という名前で絞り込みをする例です。



さらに X/Y 軸の細かい設定も行えます。



上記画像では次のような設定を行っています。

- X 軸のラベルに名前を表示
- Y 軸のラベルに同名人口数を表示
- Y 軸の最大値を 3,000,000 に変更

6.1.3 Line Chart(折れ線グラフ)の作成

さらに Line Chart を作っていきます。Apache Superset では、Line Chart は基本的に時系列であることが必須です。このため上記のテーブルや Bar Chart とは、異なるグラフを作っていきます。作成したグラフは以下です。



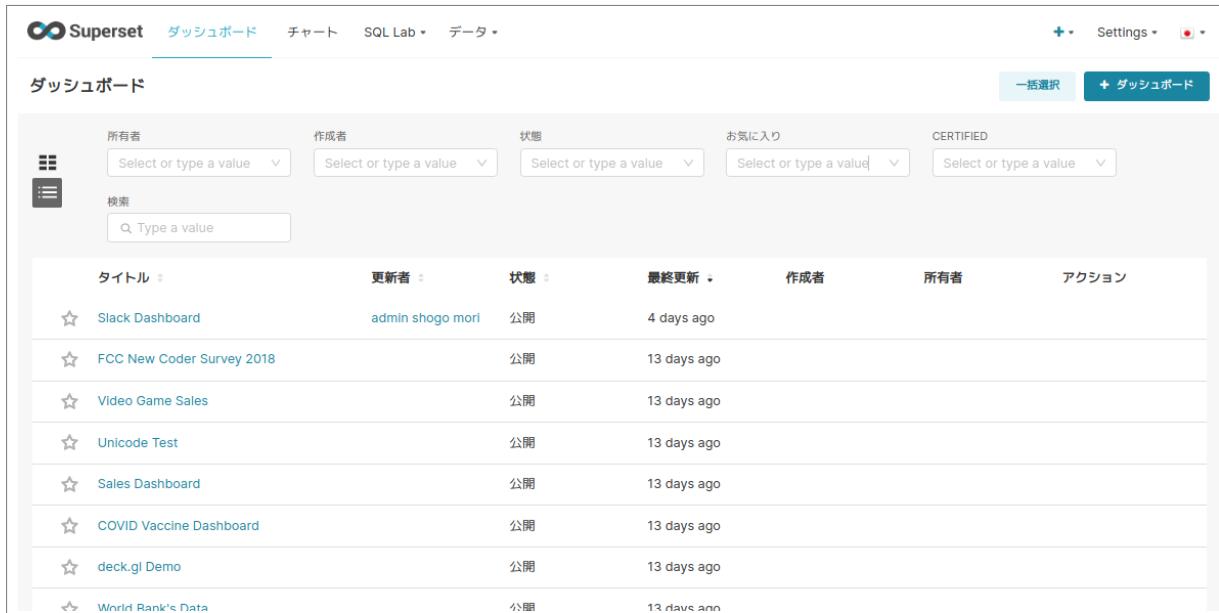
このグラフは、年代毎の名前の数の推移を表すものです。上述した通り、Line Chart は時系列になります。つまり Y 軸が時間軸で固定されます。

このためグラフの作り方としては、X 軸のデータを設定していくことになります。上記グラフの場合は、名前でグルーピングを行い、名前の総数上位 15 件のデータの推移を表示しています。

Bar Chart と同様に、X/Y 軸にラベル付をする、Y 軸の上限値を設定するなどの細かい設定も可能です。

7 ダッシュボードの管理

前章で作成したチャートは、ダッシュボードとしてまとめて閲覧が可能です。以下がダッシュボードの管理画面です。

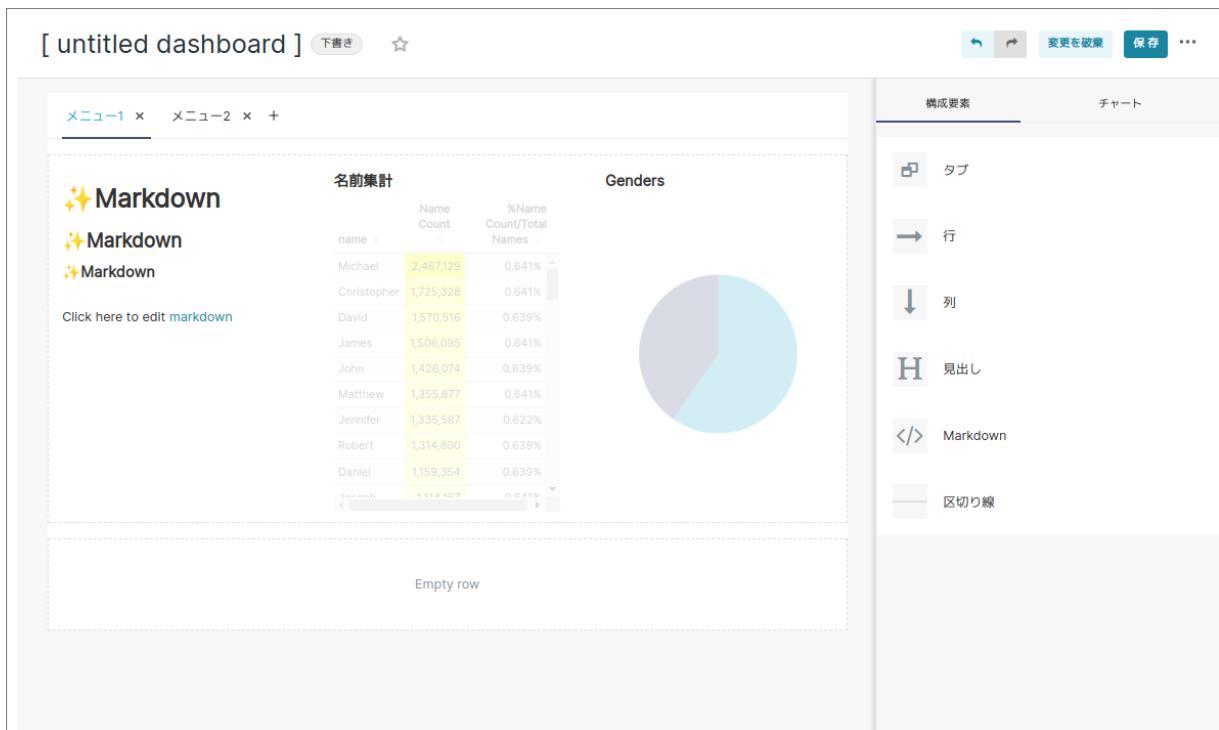


The screenshot shows the Superset dashboard management interface. At the top, there are filters for '所有者' (Owner), '作成者' (Creator), '状態' (Status), 'お気に入り' (Favorites), and 'CERTIFIED'. Below the filters is a search bar labeled '検索' (Search) with the placeholder 'Type a value'. The main area displays a table of dashboards:

| タイトル | 更新者 | 状態 | 最終更新 | 作成者 | 所有者 | アクション |
|---------------------------|------------------|----|-------------|-----|-----|--------|
| Slack Dashboard | admin shogo mori | 公開 | 4 days ago | | | [Edit] |
| FCC New Coder Survey 2018 | | 公開 | 13 days ago | | | [Edit] |
| Video Game Sales | | 公開 | 13 days ago | | | [Edit] |
| Unicode Test | | 公開 | 13 days ago | | | [Edit] |
| Sales Dashboard | | 公開 | 13 days ago | | | [Edit] |
| COVID Vaccine Dashboard | | 公開 | 13 days ago | | | [Edit] |
| deck.gl Demo | | 公開 | 13 days ago | | | [Edit] |
| World Bank's Data | | 公開 | 13 days ago | | | [Edit] |

7.1 ダッシュボードの作成

ダッシュボードの管理画面の [+ ダッシュボード] ボタンから新しいダッシュボードを作成することができます。



The screenshot shows the Superset dashboard creation interface. The dashboard is titled '[untitled dashboard]'. It contains two components:

- Markdown**: A component containing three yellow star icons and a link 'Click here to edit markdown'.
- Genders**: A chart showing the distribution of names by gender. The table data is as follows:

| Name | Count | %Name Count/Total Names |
|-------------|------------|-------------------------------|
| Michael | 2,467,129 | 0.641% |
| Christopher | 1,725,328 | 0.641% |
| David | 1,570,516 | 0.639% |
| James | 1,506,095 | 0.641% |
| John | 1,426,074 | 0.639% |
| Matthew | 1,355,877 | 0.641% |
| Jennifer | 1,335,587 | 0.622% |
| Robert | 1,314,800 | 0.639% |
| Daniel | 1,159,354 | 0.639% |
| Total | 11,114,107 | 100.00% |

The right side of the interface shows the '構成要素' (Components) panel with categories: タブ (Tab), 行 (Row), 列 (Column), 見出し (Section), and 区切り線 (Separator). There are also buttons for '変更を破棄' (Discard changes) and '保存' (Save).

ダッシュボードは、マウスのドラッグ・ドロップで構成していくことができます。構成要素には以下があります。

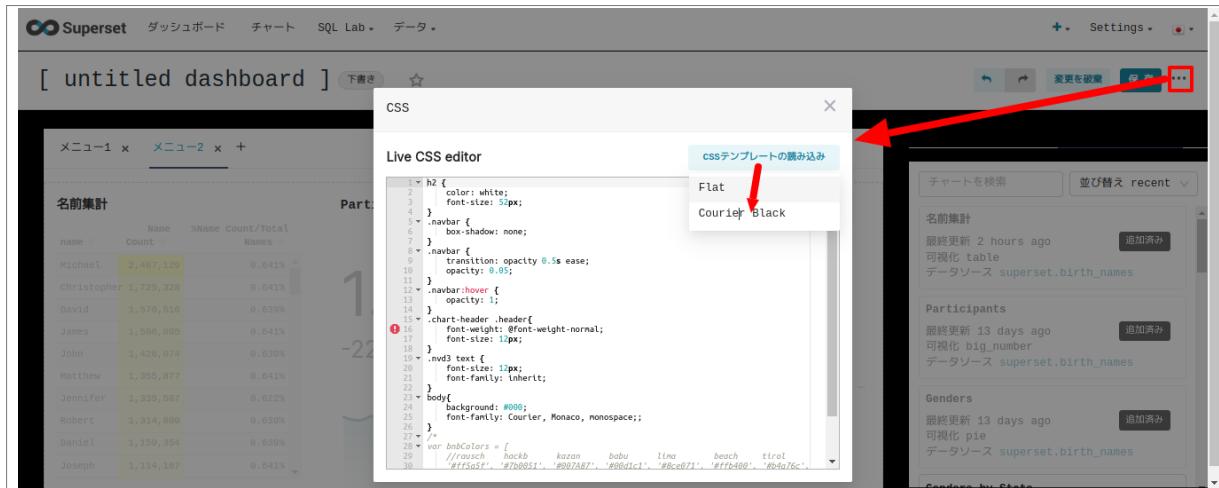
- タブ: ダッシュボード内でタブメニューの作成を行う
- 行: ダッシュボードまたはタブ内の行を作成する
- 列: ダッシュボードまたはタブ内の列を作成する
- 見出し: 見出しを表示するテキストボックスの配置
- Markdown: Markdown 形式で自由記述が可能 (HTML も利用可)
- 区切り線: 要素毎の区切り線の配置

また登録済みのチャートを選択して、表示することができます。

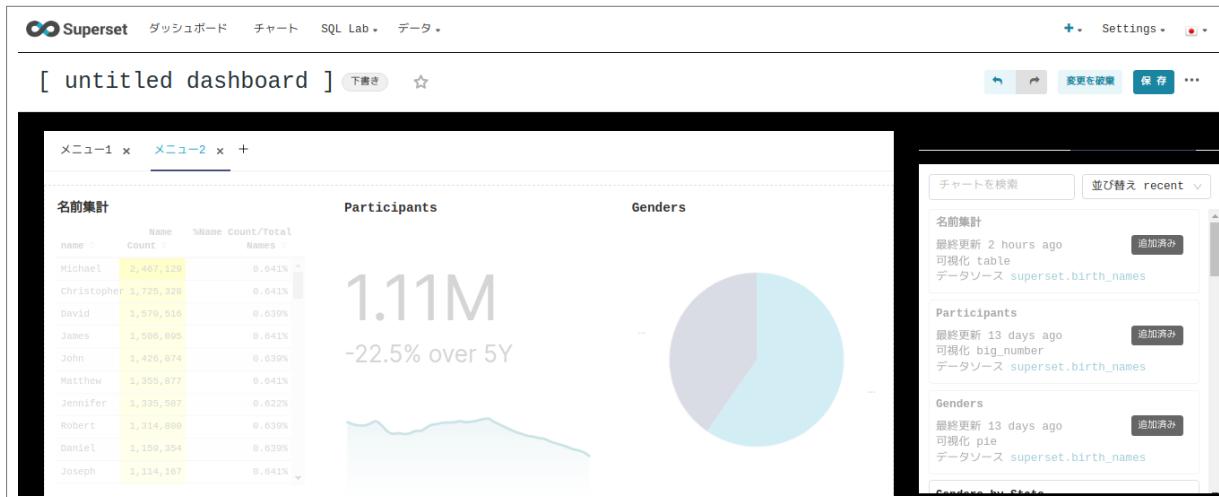
|| 7.2 ダッシュボードのデザイン変更

Apache Superset ではダッシュボード毎に CSS を登録して、色などのデザインを変更することができます。ただし HTML や javascript までは書き換えることができないため、用意されたレイアウトの中でのデザイン変更になります。

以下は、CSS の編集画面を開き、用意されている CSS テンプレートを適用している例です。テンプレートを読み込んだ後、直接書き換えることも可能です。



なお、この CSS は背景色を黒に変更する動作をするサンプルの CSS です。

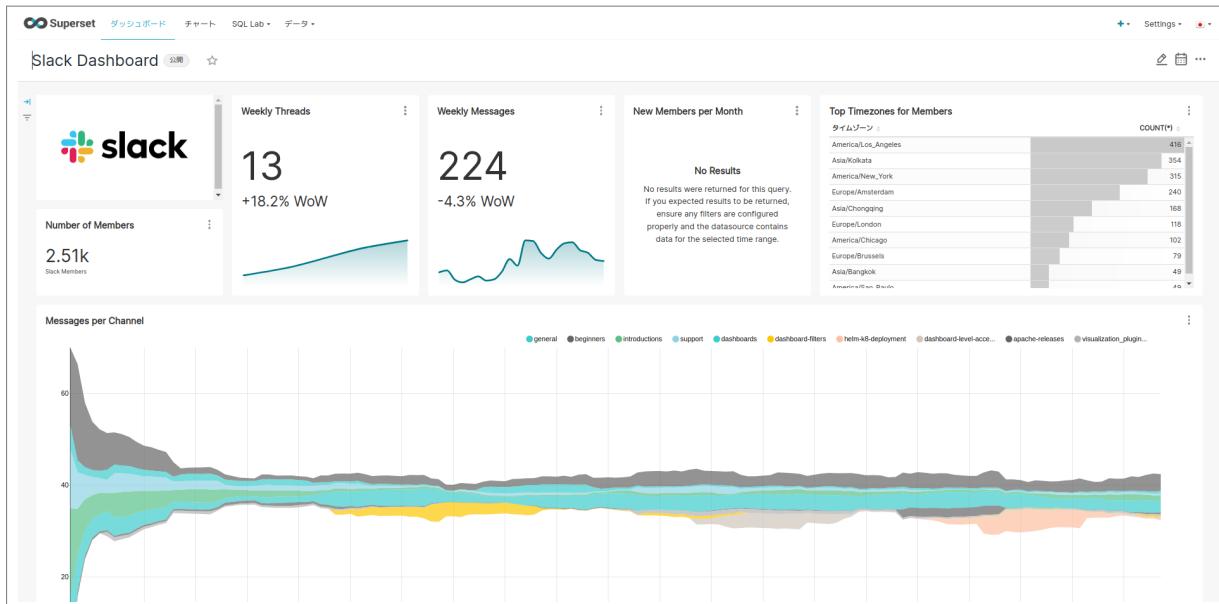


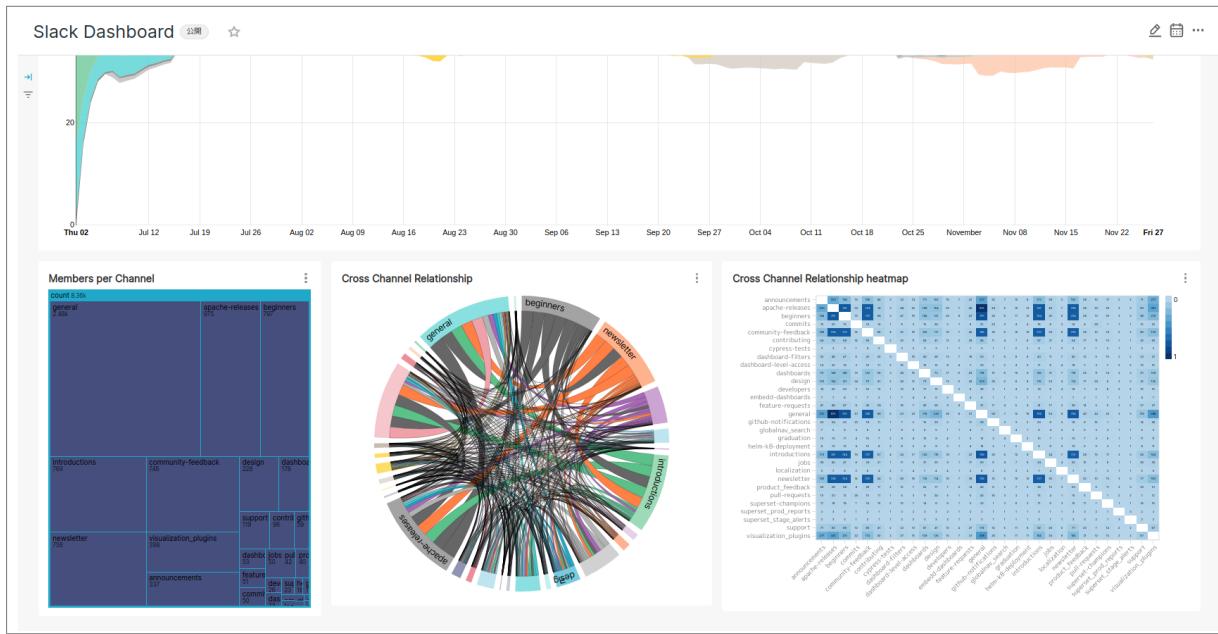
|| 7.3 ダッシュボードの例

Apache Superset のサンプルデータでは、各種ダッシュボードが予め用意されています。このダッシュボードを見ると、多彩なチャート・ダッシュボードの構成を行えることがわかります。

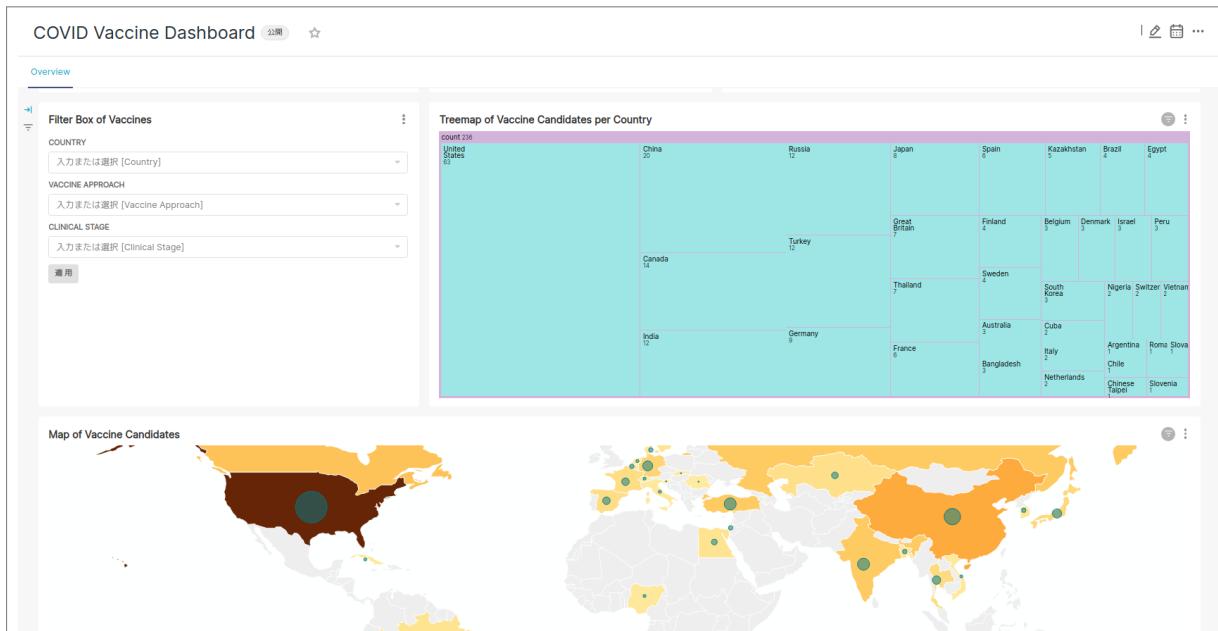
なお、表示されているデータは、あくまでサンプルですので、データ自体の信憑性については不明です。

まず Slack の利用統計情報のダッシュボードです。





次にコロナワクチンに関するダッシュボードです。



|| 7.4 ダッシュボードでの絞り込み検索

上記のコロナワクチンのダッシュボードは、ダッシュボード内に絞り込み用のフォームが用意されているのが特徴的です。

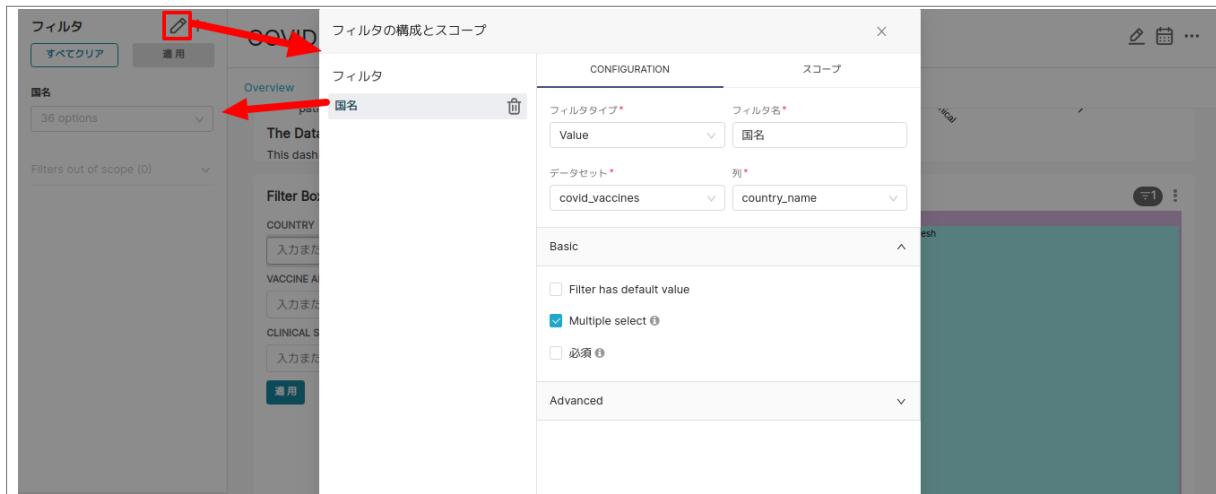
このフォームもチャートとして作成が可能になっています。

The screenshot shows the Superset interface for filtering data. On the left, there's a sidebar with a 'データセット' section containing 'superset.covid_vaccines'. Below it are sections for '指標' (Metrics) and '列' (Columns), showing various fields like 'clinical_stage', 'fda_approved_indications', and 'published_results'. In the center, a large modal window titled 'Filtering Vaccines' is open. It has three main tabs: 'COUNTRY', 'VACCINE APPROACH', and 'CLINICAL STAGE'. Under 'COUNTRY', there's a dropdown labeled '入力または選択 [Country]'. Under 'VACCINE APPROACH', there's a dropdown labeled '入力または選択 [Vaccine Approach]'. Under 'CLINICAL STAGE', there's a dropdown labeled 'country_name'. A detailed configuration panel for 'country_name' is visible, showing 'ラベル' (Label) set to 'Country', 'Default' checked, 'Sort metric' set to 'Select ...' with 'Sort ascending' checked, and several other options like '複数の選択を許可する' (Allow multiple selection) and '必須' (Required) checked. At the bottom of the configuration panel are checkboxes for 'DATE FILTER', 'INSTANT FILTERING', 'SHOW TIME GRAIN DROPDOWN', 'SHOW TIME COLUMN', and 'LIMIT SELECTOR VALUES'.

実際に絞り込みを行うと、次のように動的に画面が変化します。

The screenshot shows a dashboard titled 'COVID Vaccine Dashboard'. On the left, there's a 'Filter Box of Vaccines' section with dropdowns for 'COUNTRY' (set to 'Australia' and 'Bangladesh'), 'VACCINE APPROACH' (empty), and 'CLINICAL STAGE' (empty). Below these is a '通用' (Apply) button. To the right, there's a 'Treemap of Vaccine Candidates per Country' visualization. The treemap consists of two large teal-colored rectangles representing Australia and Bangladesh. The text 'count 6' is visible above the Australia rectangle. At the bottom of the dashboard, there's a 'Map of Vaccine Candidates' section.

またダッシュボード内にチャートとして配置せず、ダッシュボードの機能として絞り込みフォームを作ることも可能です。



|| 7.5 ダッシュボードでのクエリ確認

またダッシュボード内で、データベースへのクエリも確認することができます。前述の絞り込み検索を行った後のクエリを確認した内容が次です。

```

SELECT country_name AS country_name,
       COUNT(*) AS count
  FROM superset.covid_vaccines
 WHERE country_name IN ('Australia',
                        'Bangladesh')
 GROUP BY country_name
 LIMIT 10000;
  
```

条件に指定した内容が WHERE 句として現れています。

|| 7.6 ダッシュボードのその他機能

その他、ダッシュボードでは次のようなことが可能です。

- 一定周期での自動更新
- ダッシュボードを画像としてダウンロード
- ダッシュボードの定期レポート（後述）

|| 8 Apache Superset でのデータ操作

Apache Superset では、データベースのデータをいくつかの方法で直接操作することも可能です。

|| 8.1 データのアップロード

まず CSV によるデータアップロードです。例えば次のような CSV を用意します。

| |
|-----------------------------|
| id,name,email |
| 1,user01,user01@example.com |
| 2,user02,user02@example.com |
| 3,user03,user03@example.com |

このファイルを画面上部の [データ] の [CSV アップロード] からアップロードすることができます。



The screenshot shows the 'CSV to Database configuration' dialog in Apache Superset. It includes fields for:

- Table Name: csvtable
- CSV File: test.csv
- Database: sample
- Schema: シキーマ
- Delimiter: ,
- Table Existence: Fail
- Header Row: 0
- Index Column: Index Column

設定が必須の箇所は次の通りです。

- 新しく作成するテーブルの名前
- アップロードするファイルの選択

デフォルトの状態では、CSV ファイルの 1 行目がヘッダ行として扱われ、ヘッダの値に沿ってテーブルのカラムが作成されます。

カラムの型は自動的に判断して設定されます。

|| 8.2 SQL Lab でのデータ操作

Apache Superset では、SQL Lab という SQL クエリを発行するためのインターフェースがあります。次の画像は、上記の CSV アップロードで作成した、csvtable を表示している例です。テーブルを選択した時点で、テーブルの情報が表示されます。

The screenshot shows the Apache Superset interface. In the top navigation bar, 'Superset' is selected. Below it, there are tabs for 'ダッシュボード', 'チャート', 'SQL Lab', and 'データ'. A search bar at the top right contains the text 'CHAPTER 8. APACHE SUPERSET でのデータ操作'. On the left, a sidebar displays database schema information for 'examples' and 'sample' databases, including tables like 'weather_info', 'a', and 'birth_names'. The main area shows a preview of a CSV table named 'csvtable'. The preview table has three rows with columns 'id', 'name', and 'email'. The data is as follows:

| | id | name | email |
|---|--------|------|--------------------|
| 1 | user01 | | user01@example.com |
| 2 | user02 | | user02@example.com |
| 3 | user03 | | user03@example.com |

さらに SQL を実行することもできます。

The screenshot shows the Apache Superset interface with a SQL query editor. The query is: `select * from csvtable where name='user01'`. The results show one row returned, which is the same as the preview in the previous screenshot. The data is as follows:

| | id | name | email |
|---|--------|------|--------------------|
| 1 | user01 | | user01@example.com |

この機能を使うことで、データベースのデータを直接検索したり、時にはデータを更新することも可能です。

なお更新系の操作はデータベース登録時の設定で、更新系クエリを許可する必要があります。

9 Apache Superset のユーザと権限管理

Apache Superset は、ユーザを複数作成してユーザ毎に操作・閲覧権限を割り当てることができます。

9.1 ユーザの管理

Apache Superset のユーザ管理画面は、右上の Settings メニューから移動できます。



The screenshot shows the Apache Superset User Management interface. On the left, there is a table listing two users: 'admin' and 'テスト'. The 'admin' user has the name 'shogo mori', username 'admin', email 'smori@designet.co.jp', and is marked as 'True' for active status. The 'テスト' user has the name 'ユーザ', username 'test', email 'tamura@designet.co.jp', and is also marked as 'True'. On the right, a vertical sidebar menu is open under the 'セキュリティ' section, showing options like 'ユーザー一覧', 'ロール一覧', and '操作履歴'. Other sections like 'CSSテンプレート' and 'アラートとレポート' are also visible.

この画面からユーザ追加や削除・編集の処理を行うことができます。以下はユーザの追加画面です。



The screenshot shows the Apache Superset User Addition form. It consists of several input fields with validation requirements indicated by red asterisks (*). The fields are:

- 名前(名) ***: A text input field labeled '名前(名)' with placeholder text 'ユーザーの名前を入力して下さい'.
- 名字(姓) ***: A text input field labeled '名字(姓)' with placeholder text 'ユーザーの姓を入力して下さい'.
- ユーザー名 ***: A text input field labeled 'ユーザー名' with placeholder text 'ユーザー名はデータベース又は LDAP 上での認証のために有効です。OID認証では無視されます'.
- 有効?**: A checkbox labeled '有効?' with placeholder text 'ユーザーの削除は良いポリシーではありません。ユーザーを無効にする事をお薦めします'.
- メール ***: A text input field labeled 'メール' with placeholder text 'これはユーザーのメールです。OID認証でも利用されます'.
- 役割**: A dropdown menu labeled 'Select Value' with placeholder text 'このアプリケーション上のユーザーロールです。ロールに伴う「許可」リストは附属されます'.

|| 9.2 権限の管理

Apache Superset では、非常に細かい権限が用意されており、それらを組み合わせたロールを作成することができます。ロールはユーザに割り当てて、ユーザの操作を制限することができます。

プリセットのロールとして次が用意されています。

- Admin
 - 管理者用の全権限を持つロール
- Alpha
 - 全てのデータにアクセスできるが、他のユーザの制御はできないロール
- Gamma
 - 許可されたデータソースから作成されたダッシュボードやチャートを操作できる権限
 - いわゆる一般ユーザを想定しているロール
 - データソースの許可は別途追加が必要
- Public
 - ログインしていないユーザに割り当てられるロール
 - 内部設定ファイルの「PUBLIC_ROLE_LIKE」の値に登録されたロールと同じ動作となる
 - 例: PUBLIC_ROLE_LIKE=Gamma の場合、ログインせず Gamma ロールと同じ操作が可能

このプリセットのロールの設定を確認すると、次のように大量の権限が設定されていることがわかります。

The screenshot shows the Apache Superset interface for managing roles. The top navigation bar includes links for 'Dashboard', 'Charts', 'SQL Lab', 'Data', and 'Settings'. The main title is 'ロールを編集' (Edit Role). Below it, there are tabs for 'Detail' and 'ユーザー一覧' (User List), with 'Detail' currently selected. The 'Name' field contains 'Gamma'. The 'Permissions' section lists numerous permissions, many of which are checked (indicated by a red asterisk). Some examples of the permissions listed include: can read on SavedQuery, can read on CssTemplate, can read on ReportSchedule, can read on Chart, can read on Annotation, can read on Dataset, can read on Dashboard, can read on Database, can read on Query, can read on SecurityRestApi, can write on SavedQuery, can write on CssTemplate, can write on ReportSchedule, can write on Chart, can write on Annotation, can write on Dashboard, can this form post on ResetMyPasswordView, can this form post on UserInfoEditView, can this form post on CsvToDatabaseView, can this form post on ExcelToDatabaseView, can this form post on ColumnarToDatabaseView, can this form get on ResetMyPasswordView, can this form get on UserInfoEditView, can this form get on CsvToDatabaseView, can this form get on ExcelToDatabaseView, can this form get on ColumnarToDatabaseView, can userinfo on UserDBModelView, can list on AsyncEventsRestApi, can list on FilterSets, can list on DynamicPlugin, can list on DashboardEmailScheduleView, can list on SliceEmailScheduleView, can list on AlertModelView, can list on AlertLogModelView, can list on AlertObservationModelView, can list on DruidDatasourceModelView, can list on DruidClusterModelView, can list on DruidMetricInlineView, can list on DruidColumnInlineView, can add on FilterSets, can add on DashboardEmailScheduleView, can add on SliceEmailScheduleView, can add on AlertModelView, can show on SwaggerView, can show on DynamicPlugin, can show on DashboardEmailScheduleView, can show on SliceEmailScheduleView, can show on AlertModelView, can show on AlertLogModelView, can show on AlertObservationModelView, can show on DruidDatasourceModelView, can show on DruidClusterModelView, can edit on FilterSets, can edit on DashboardEmailScheduleView, can edit on SliceEmailScheduleView, can edit on AlertModelView, can delete on FilterSets, can delete on TableSchemaView, can delete on TabStateView, can delete on TagView, can delete on DashboardEmailScheduleView, can delete on SliceEmailScheduleView, can delete on AlertModelView, can delete on UserDBModelView, can get on OpenApi, can get on MenuApi.

画像は途中で切っていますが、さらに画面末尾まで権限の設定が続いています。現在全体では約 230 件の権限が用意されています。

これらの権限の詳細については、残念ながら現在はマニュアル化されていません。権限のキーワードからどのような動作かを想像するしかない状態です。

このため設定の基本としては Gamma ロールを複製して、新たなロールを作っていく方法が最も簡単な方法です。

9.3 最小限のロールの作成検証

Gamma ロールもある程度制限されているとはいえ、大量の許可が行われていることには違いないため、単純にダッシュボードの表示のみを許可する必要最小限のロールを作成しました。

The screenshot shows the Apache Superset interface for managing roles. A role named 'myrole' is selected. The 'Detail' tab is active, showing the role's name and its assigned permissions. The 'User' tab lists a single user, 'テストユーザ'. The 'Permissions' section contains several checkboxes, many of which are checked, indicating specific actions like reading on various components and dashboard access.

上記のロールは、動作させた内容からの推測ではそれぞれ次のような意味を持ちます。

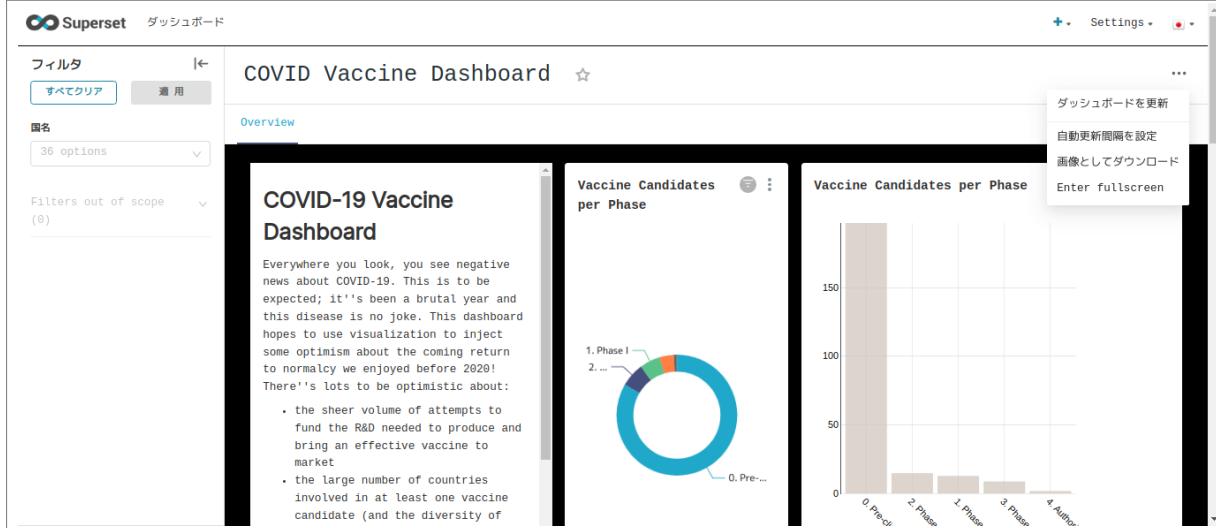
- can dashboard on Superset
 - ダッシュボードの利用権限
- can read on Dashboard
 - ダッシュボードの参照権限
- can explore on Superset
 - データの検索権限
- can explore json on Superset
 - json データの検索権限
- can read on Chart
 - チャートの参照権限
- menu access on Dashboards
 - 画面上部のダッシュボードメニューの表示
- datasource access on [examples].[covid_vaccines](id:13)
 - covid_vaccines データセットへのアクセス権限
 - 利用するデータセット毎に追加が必要

ここまでが必須の権限です。これらの権限だけでも利用できますが、画面に大量のエラーが発生します。エラー抑制のため、次の権限も追加が必要です。

- can read on SavedQuery
 - 保存クエリの参照権限
- can read on CssTemplate

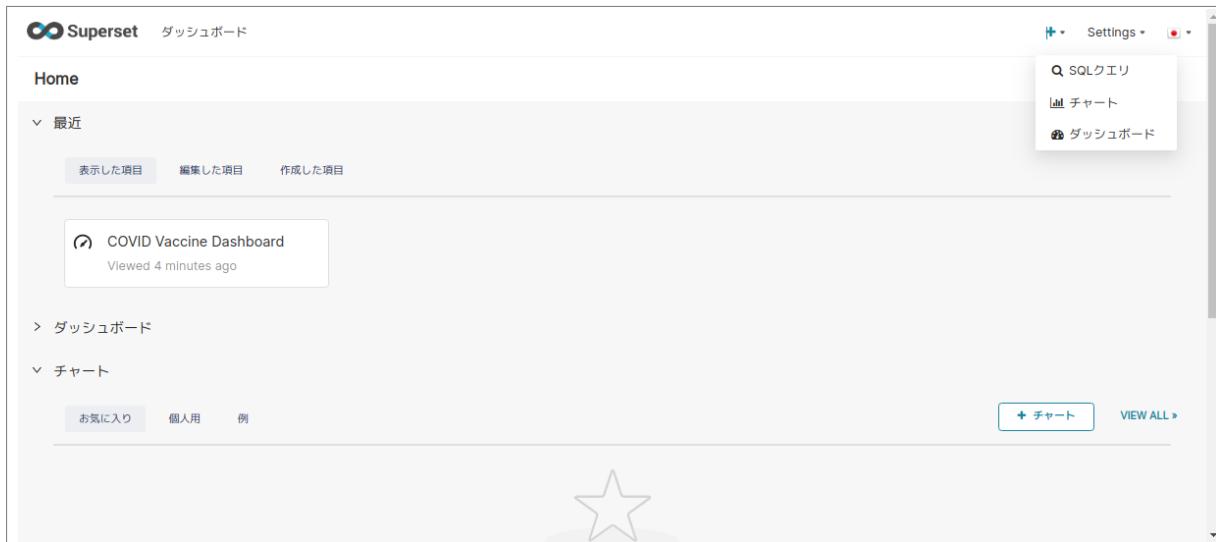
- CSS テンプレートの読み込み権限
- can recent activity on Superset
 - 最近アクティビティの利用権限
- can favstar on Superset
 - お気に入り (ダッシュボード等) 情報の利用権限

これらの許可をしたダッシュボードは次のように見えます。



上部のメニューがダッシュボードのみになり、またダッシュボードの編集などができなくなっています。

またホーム画面は次のように権限のあるダッシュボードだけが見える状態になっています。



ただ、権限を最小限にした状態でも、右上のメニューを消すことはできませんでした。

SQL クエリやチャートなどのリンクは使える状態です。ただ、実際に遷移すると権限のエラーになりホームに戻される動作となるため、実用上セキュリティの問題はありませんでした。

このように権限を組み合わせることで、特定の人に特定のダッシュボードを見せるということが可能であることがわかりました。

| 10 発展的な設定

|| 10.1 非同期処理の実行設定

Aapache Superset には大量のデータを扱うようなケースを想定した設定方法が存在しています。上記の方法でも基本的に動作しますが、DB の検索などサーバ側の処理で遅延が生じた場合、接続がタイムアウトするケースがあります。このようなケースに対応するために、非同期な処理を実現するための構成が用意されています。

非同期処理を有効化すると次の機能の利用が可能になります。

- WEB サーバではないワーカーでの DB リクエストの処理
- スケジュールレポート（メール等でダッシュボードの画像を自動通知する）

|| 10.2 非同期処理関連機能の設定

||| 10.2.1 コンポーネントのインストール

必要になるコンポーネントは以下です。

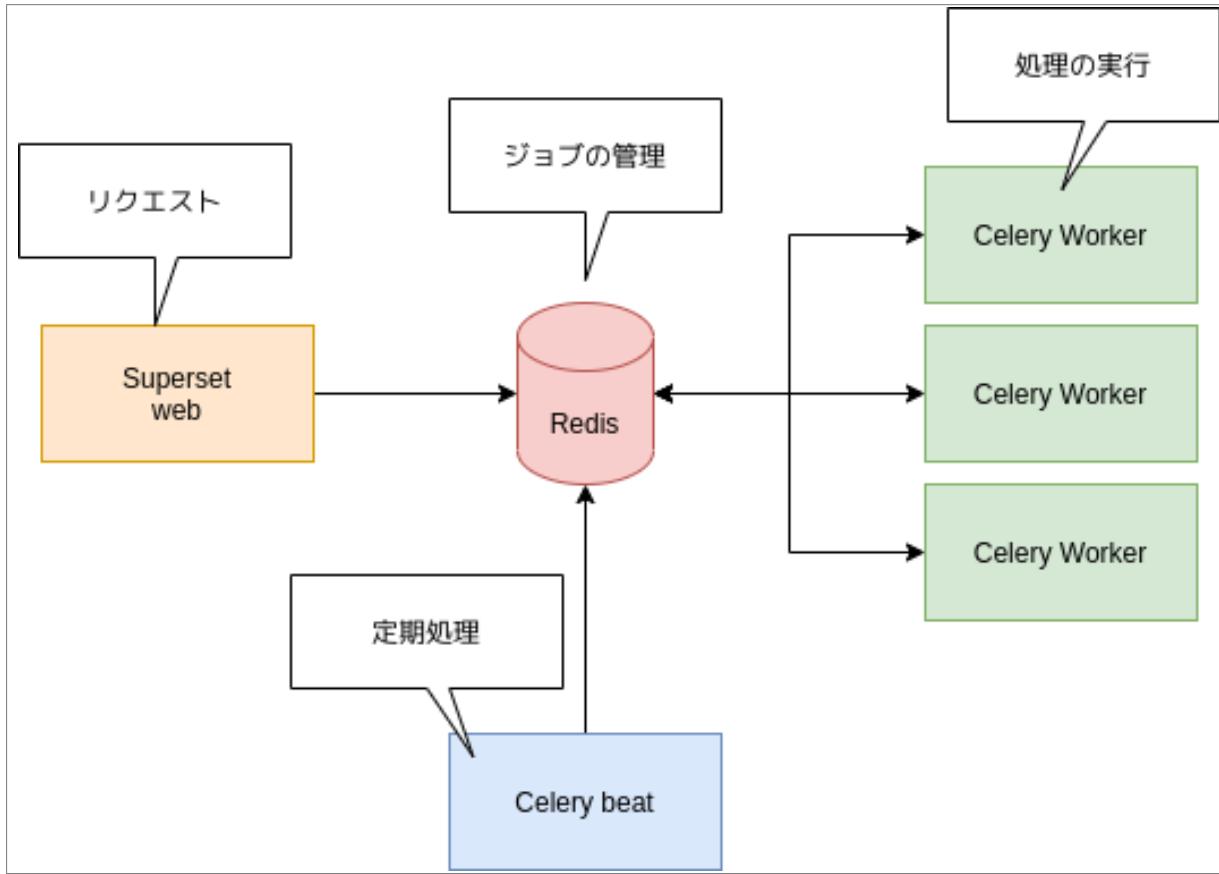
- Celery
 - Python でメッセージキューの管理・タスクの実行を行うための Python ライブライ
- Redis
 - Celery のバックエンドデータストア
- Chrome/web ドライバ

Redis は基本的なインストールの際にも、キャッシュ用途でインストールするため、追加のインストールは不要です。また Celery も Apache Superset を pip コマンドでインストール際に自動的にインストールされます。

Chrome と WEB ドライバは、スクリーンショットをメールで通知する機能を利用する場合に必要です。

||| 10.2.2 動作イメージ

Celery を使った構成の動作イメージは次の通りです。



10.2.3 Chrome と WEB ドライバのインストール

Chrome と Web ドライバは次のようにインストールします。

```
# Chrome のインストール
$ sudo sh -c 'echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/
-sources.list.d/google-chrome.list'
$ sudo apt update
$ sudo apt install google-chrome-stable
$ curl -LO https://chromedriver.storage.googleapis.com/97.0.4692.71/chromedriver_linux64.zip

# web ドライバのインストール
$ unzip chromedriver_linux64.zip
$ sudo mv chromedriver /usr/local/bin/
$ sudo chmod a+x /usr/local/bin/chromedriver
```

なお Web ドライバは Chrome のバージョンと合わせてインストールしてください。

III 10.2.4 追加設定

またこれらのコンポーネントを使うためには、設定の追加が必要です。次のように設定を追加します。

```
FLASK_APP = 'superset'
ROW_LIMIT = 5000
SECRET_KEY = 'my_secret_key'
# SQLALCHEMY_DATABASE_URI = 'sqlite:///opt/superset/.superset/superset.db'
SQLALCHEMY_DATABASE_URI = 'mysql://superset:supersetPassword@127.0.0.1/superset'

WTF_CSRF_ENABLED = True
WTF_CSRF_EXEMPT_LIST = []
WTF_CSRF_TIME_LIMIT = 60 * 60 * 24 * 365

CACHE_CONFIG = {
    'CACHE_TYPE': 'redis',
    'CACHE_DEFAULT_TIMEOUT': 60 * 60 * 24, # 1 day default (in secs)
    'CACHE_KEY_PREFIX': 'superset_cache',
    'CACHE_REDIS_PORT': 6379,
    'CACHE_REDIS_DB': 1,
    'CACHE_REDIS_URL': 'redis://127.0.0.1:6379/0',
}

DATA_CACHE_CONFIG = {
    **CACHE_CONFIG,
    "CACHE_KEY_PREFIX": "superset_data_cache",
}

LANGUAGES = {
    "en": {"flag": "us", "name": "English"},
    "ja": {"flag": "jp", "name": "Japanese"},
}

# レポート機能の有効化
FEATURE_FLAGS = {
    "ALERT_REPORTS": True,
}

# 非同期関連設定
from cachelib.redis import RedisCache
RESULTS_BACKEND = RedisCache(
    host='127.0.0.1', port=6379, key_prefix='superset_results')

from celery.schedules import crontab
class CeleryConfig:
    BROKER_URL = 'redis://127.0.0.1:6379/0'
    CELERY_IMPORTS = ('superset.sql_lab', "superset.tasks", "superset.tasks-thumbnails", )
    CELERY_RESULT_BACKEND = 'redis://127.0.0.1:6379/0'
    CELERYD_PREFETCH_MULTIPLIER = 10
    CELERY_ACKS_LATE = True
    CELERY_ANNOTATIONS = {
```

```
'sql_lab.get_sql_results': {
    'rate_limit': '100/s',
},
'email_reports.send': {
    'rate_limit': '1/s',
    'time_limit': 600,
    'soft_time_limit': 600,
    'ignore_result': True,
},
}
CELERYBEAT_SCHEDULE = {
'reports.scheduler': {
    'task': 'reports.scheduler',
    'schedule': crontab(minute="*", hour="*"),
},
'reports.prune_log': {
    'task': 'reports.prune_log',
    'schedule': crontab(minute=0, hour=0),
},
}
}

CELERY_CONFIG = CeleryConfig

# ダッシュボードのスクリーンショット取得時の待ち時間
SCREENSHOT_LOCATE_WAIT = 100
SCREENSHOT_LOAD_WAIT = 600

# レポートメールを送信する場合の設定
SMTP_HOST = "192.168.100.10" #change to your host
SMTP_PORT = 25 #
SMTP_STARTTLS = False
SMTP_SSL = False
SMTP_USER = ""
SMTP_PASSWORD = ""
SMTP_MAIL_FROM = "user@example.com"

# スクリーンショット取得のための web ドライバの設定
WEBDRIVER_TYPE = "chrome"
WEBDRIVER_OPTION_ARGS = [
    "--force-device-scale-factor=2.0",
    "--high-dpi-support=2.0",
    "--headless",
    "--disable-gpu",
    "--disable-dev-shm-usage",
    "--no-sandbox",
    "--disable-setuid-sandbox",
    "--disable-extensions",
]
WEBDRIVER_BASEURL="http://127.0.0.1:8088"
WEBDRIVER_BASEURL_USER_FRIENDLY="http://127.0.0.1:8088"
```

なお Apache Superset の設定は、マニュアル化されていないものが多数存在します。設定の詳細は、次

の設定に関連する Python コードのコメントを読む必要があります。

- <https://github.com/apache/superset/blob/master/superset/config.py>

III 10.2.5 サービスの起動

非同期設定を有効にするために、celery 関連のサービスを起動します。

まず Celery beat を起動します。Celery beat はタスクの定期・スケジュール実行のためのサービスです。

```
$ export SUPERSET_CONFIG_PATH=/opt/superset/etc/superset_config.py
$ export VIRTUAL_ENV=/opt/superset/python-superset
$ /opt/superset/python-superset/bin/celery --app=superset.tasks.celery_app:app beat --pidfile /
  ↳tmp/celerybeat.pid --schedule /tmp/celerybeat-schedule
```

次に Celery worker を起動します。これはサーバサイドの処理を実行するサービスです。

```
$ export SUPERSET_CONFIG_PATH=/opt/superset/etc/superset_config.py
$ export VIRTUAL_ENV=/opt/superset/python-superset
$ /opt/superset/python-superset/bin/celery --app=superset.tasks.celery_app:app worker --
  ↳pool=prefork -O fair -c 4
```

なお systemd に登録する場合、ユニットファイルは次のようになります。

- /etc/systemd/system/superset-beat.service

```
[Unit]
Description=Superset celery beat
After=network.target

[Install]
WantedBy=multi-user.target

[Service]
Type=simple
User=superset
Group=superset
WorkingDirectory=/opt/superset
Environment=SUPERSET_CONFIG_PATH=/opt/superset/etc/superset_config.py
Environment=VIRTUAL_ENV=/opt/superset/python-superset
Environment=PATH=/opt/superset/python-superset/bin
ExecStart=/opt/superset/python-superset/bin/celery --app=superset.tasks.celery_app:app beat --
  ↳pidfile /tmp/celerybeat.pid --schedule /tmp/celerybeat-schedule
```

- /etc/systemd/system/superset-worker.service

```
Description=Superset celery worker
After=network.target

[Install]
WantedBy=multi-user.target

[Service]
```

```
Type=simple
User=superset
Group=superset
WorkingDirectory=/opt/superset
Environment=SUPERSET_CONFIG_PATH=/opt/superset/etc/superset_config.py
Environment=VIRTUAL_ENV=/opt/superset/python-superset
Environment=PATH=/opt/superset/python-superset/bin
ExecStart=/opt/superset/python-superset/bin/celery --app=superset.tasks.celery_app:app worker -
  -c 4
  -pool=prefork -O fair -c 4
```

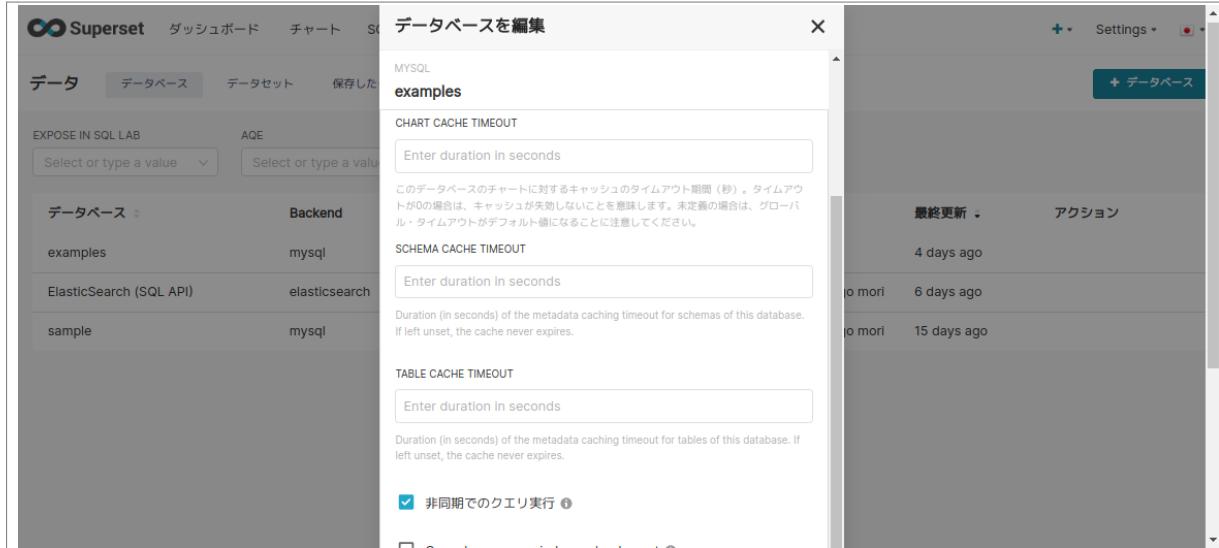
なお superset のサービスの再起動も必要です。

|| 10.3 非同期処理有効環境で利用可能な機能

|| 10.3.1 サーバサイドクエリ

非同期処理が有効な場合、Apache Superset が動作する WEB サーバではなく、Celery ワーカーで DB へのリクエストを処理できるようになります。

この動作は、非同期処理を有効にしても、自動では有効になりません。データベース毎に次の設定を有効化する必要があります。



この設定を追加すると、DB への検索などのリクエストが Celery ワーカーで処理されるようになります。

|| 10.3.2 スケジュールレポート

レポート機能を有効化することで、ダッシュボードにレポート設定が表示されるようになります。

The screenshot shows the Superset interface with the 'COVID Vaccine Dashboard' selected. A modal window titled 'Edit Email Report' is open. Inside the modal, there's a 'REPORT NAME' field containing 'Report', a 'DESCRIPTION' field with the placeholder 'Include a description that will be sent with your report', and a 'Schedule' section. The 'Schedule' section contains dropdowns for 'Every' (set to '日'), 'at' (set to '17 : 45'), and 'TIMEZONE' (set to 'GMT +09:00 (Asia/Jayapura)'). At the bottom of the modal are 'キャンセル' and '保存' buttons. A red arrow points from the 'Schedule' section towards the '保存' button. In the background, a chart titled 'Vaccine Candidates per Phase' is visible, showing a bar chart with categories 0. Pre-clinical, 1. Phase I, 2. Phase II or Combined, 3. Phase III, and 4. Authorized.

レポートを通知するスケジュールを入力することで、メールでダッシュボードのキャプチャ画像を送ることができます。

以下が実際のメールです。

The screenshot shows an email message with the subject '[Report] Report: COVID Vaccine Dashboard' and the date '2022/01/26 17:45'. The email body displays the 'COVID-19 Vaccine Dashboard' from Superset. It includes the main dashboard content, which features a circular chart labeled 'Vaccine Candidates per Phase' and a bar chart also labeled 'Vaccine Candidates per Phase'. Below the charts, there is a section titled 'The Dataset' with a note about the dataset being powered by the Milken Institute. At the bottom of the email body, there is a 'Filter Box of Vaccines' and a 'Treemap of Vaccine Candidates per Country' section.

この機能を使うことで、ダッシュボードにアクセスせず、定期的にダッシュボードを閲覧することができます。

| 11 Apache Superset まとめ

|| 11.1 Apache Superset の特徴

今回の調査にて、Apache Superset には次のような特徴があることがわかりました。

- 多彩な可視化表現を実装している
- RDBMS 互換の多くのデータベースに対応している
- 綺麗なインターフェースでデータの参照が可能
- 新たなデータを登録して検索対象にできる
- SQL クエリを直接実行することが可能
- 日本語対応が可能 (現在は翻訳途中の状態)
- 非常に細かい権限管理が可能
- ダッシュボードを定期的にレポートすることができる

特に OSS の BI ツールでは、日本語に対応しているものは稀なため、日本人にとって使いやすいツールの一つです。

|| 11.2 Apache Superset の問題点

||| 11.2.1 要求されるスキルレベルが高い

基本的に SQL や扱うデータベースに熟知していることが求められるツールです。このため、データの管理を行ったりダッシュボードを作る人間には、比較的高いスキルが求められます。

ただ、権限管理で一般的な人にダッシュボードの参照だけを許可することもできるため、利用の棲み分けは充分可能です。

||| 11.2.2 データベースの対応

対応するデータベースが RDBMS に限られるため、MongoDB などの NoSQL 系のデータベースには対応することができません。

||| 11.2.3 日本語の問題

日本語化がまだ行われていない部分が多く、今後の翻訳作業を待つ必要があります。また一部、チャートのラベルなど一部日本語の名前を登録するとエラーが発生する箇所があり、全てで日本語が使えるわけではありません。

|| 11.2.4 権限の設定

権限設定毎にどういう挙動が許可されるのかマニュアル化されていません。現在は、権限の名前と動作、またはソースコードから推測が必要です、

|| 11.3 まとめ

Apache Superset は、RDBMS 系のデータベースを利用している企業・団体であれば、非常に有効に利用できるツールであるということがわかりました。

今後は、日本語化の対応や権限の詳細調査などを行っていきたいと考えています。