

SQL Stream Overview

Date published: 2019-12-17

Date modified: 2021-03-25



Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Introduction to SQL Stream Builder.....	4
Key features of SSB.....	4
SQL Stream Builder Architecture.....	5

Introduction to SQL Stream Builder

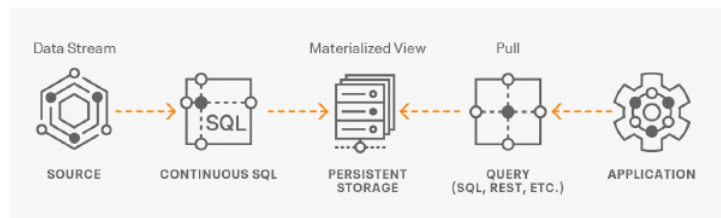
Cloudera Streaming Analytics offers an easy to use and interactive SQL Stream Builder as a service to create queries on streams of data through SQL.

The SQL Stream Builder (SSB) is a comprehensive interface for creating stateful stream processing jobs using SQL. By using SQL, you can simply and easily declare expressions that filter, aggregate, route, and otherwise mutate streams of data. SSB is a job management interface to compose and execute Continuous SQL on streams, as well as to create durable data APIs for the results.

SSB runs in an interactive fashion where you can quickly see the results of your query and iterate on your SQL syntax. The executed SQL queries run as jobs on the Flink cluster, operating on boundless streams of data until cancelled. This allows you to author, launch, and monitor stream processing jobs within SSB as every SQL query is a Flink job.

What is Continuous SQL?

Continuous SQL is using Structured Query Language (SQL) to create computations against unbounded streams of data, and show the results in a persistent storage. The result stored in a persistent storage can be connected to other applications to have an analytical visualization of your data. Compared to traditional SQL, in Continuous SQL the data has a start, but no end. This means that queries continuously process results to a sink or other target types. When you define your job in SQL, the SQL statement is interpreted and validated against a schema. After the statement is executed, the results that match the criteria are continuously returned.



Key features of SSB

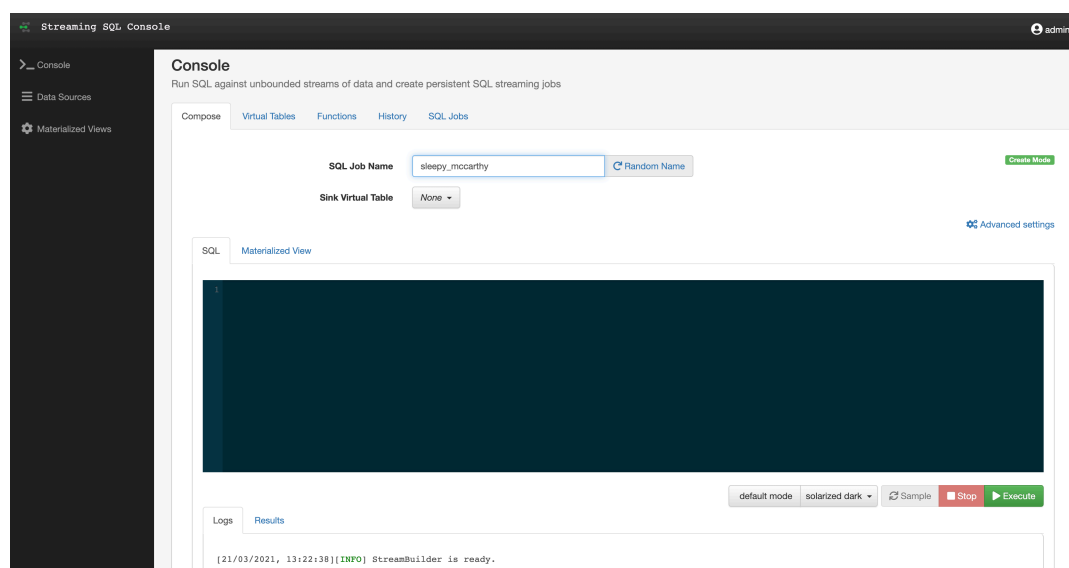
SQL Stream Builder (SSB) within Cloudera supports out-of-box integration with Flink, Kafka as virtual table sink and source. For integration with Business Intelligence tools you can create Materialized Views.

Integration with Flink

With the integration of Flink, you can use and submit Flink jobs without using Java as SSB automatically builds and runs the Flink job in the background.

Streaming SQL Console

SSB comes with a User Interface that allows you to easily create and manage your SQL jobs in one place.



Virtual Tables

SSB processes data from a source to a sink using what you specify in the SQL query. You can also show your result in the web browser. When a source or sink is created, you assign it a virtual table name. This virtual table name is used to address the FROM table in your query (source), and specify the destination (sink) in the interface. This allows you to create powerful aggregations, filters or any other SQL expressions against the stream.

Materialized Views

SSB has the capability to materialize results from a Streaming SQL query to a persistent view of the data that can be read through REST. Applications can use this mechanism to query streams of data without deploying database systems. Materialized Views are built into the SQL Stream Builder service, and require no configuration or maintenance. The Materialized Views act like a special kind of sink, and can even be used in place of a sink.

Detect Schema

SSB is capable of reading messages in a topic, identifying their data structure and sampling the schema to the UI. This is a useful function when you do not use Schema Registry.

Input Transform

In case you are not aware of the incoming data structure or raw data is collected from sensors, you can use the Input Transform to clean up and organize it before querying.

User Defined Functions

You can create customized and complex SQL queries by using User Defined Functions to enrich your data, apply computations or a business logic on it.

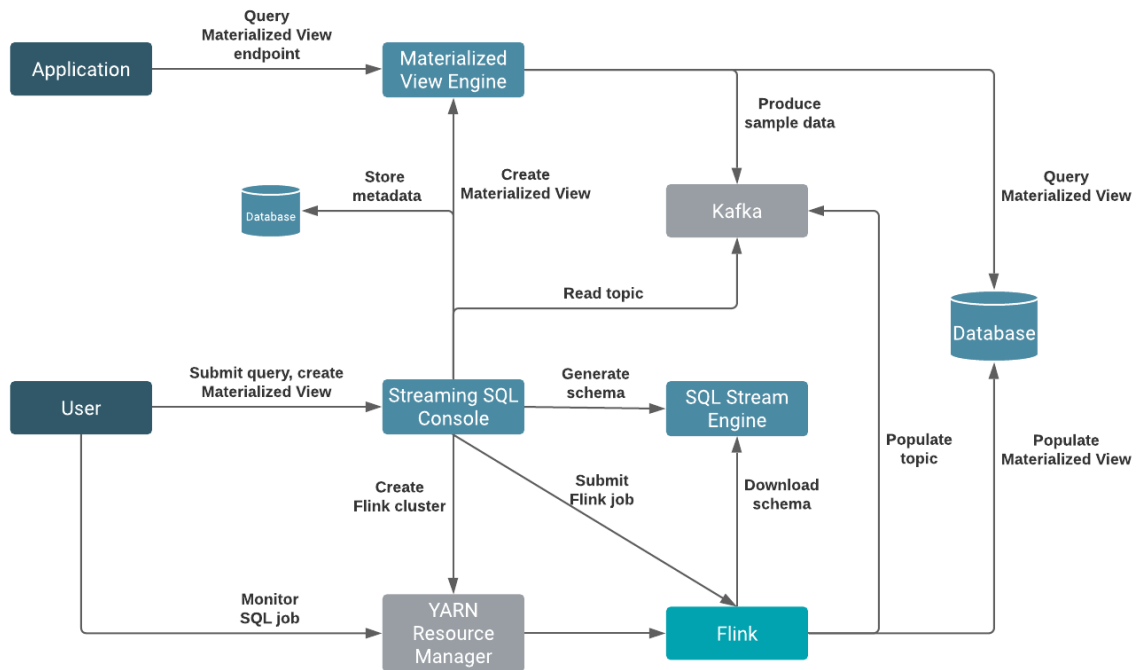
SQL Stream Builder Architecture

The SBB service is integrated on the Cloudera platform connected to Flink and its services: YARN, Kafka and Schema Registry. The primary point of user interaction for SQL Stream Builder is the Console component. When you submit a query using the UI, a Flink job is created on the cluster. The schema corresponding to the query is downloaded through Schema Registry. The Kafka topic is also populated by the Flink job submission. You are able to monitor and manage your Flink jobs using the YARN Resource Manager or the Flink Dashboard.

SSB consists of the following main components:

- SQL Stream Engine

- Streaming SQL Console
- Materialized View Engine



The primary point of user interaction for SQL Stream Builder is the Console component. When you submit a query using the Streaming SQL Console, a Flink job is automatically created in the background on the cluster. SSB also requires a Kafka service on the same cluster. This mandatory Kafka service is used to automatically populate topics for the websocket output. The websocket output is needed for sampling data to the Console, and when no virtual table sink is added to the SQL query.

When a Materialized View query is submitted, Flink generates the data to the Materialized View database from which the Materialized View Engine queries the required data. The Streaming SQL Console and the Materialized Views need databases where the metadata of SQL jobs are stored and from which the Materialized View Engine queries data to create the views. SSB supports MySQL/ MariaDB and PostgreSQL as databases. For the Streaming SQL Console, you can choose MySQL/MariaDB or PostgreSQL. However, you must install PostgreSQL to be able to create Materialized Views.