

# CLUSTERA SQL STREAM BUILDER

Democratizing real-time data access with SQL



Why Cloudera Streaming Analytics?

- **Next-generation stream processing engine** - Can process hundred billion events per day, consuming both offline and live data, and continuously monitoring and processing events in real-time.
- **Stream processing simplified** - Stream building is no longer for just the Java and Scala developers but for query builders who are more familiar with SQL.
- **Multi-Cloud and Hybrid Cloud Strategy** - With the power of CDP, customers can choose a flexible deployment model for your streaming architecture that spans across edge, on-premises and multi-cloud environments.
- **Unified Streaming and Batch** - The stream processing engine can join and enrich from a variety of sources including streaming sources like Apache Kafka as well as batch sources like Hive, Kudu and JDBC.

Stream Processing is a critical function of any streaming architecture to handle massive volumes of high velocity data streams in real-time. Cloudera SQL Stream Builder, the latest Cloudera Streaming Analytics component, simplifies streaming application development, accelerates time to market, and democratizes access to real-time data using continuous SQL. It creates a rich iterative development environment and robust management framework using Apache Flink via industry-standard SQL.

## Challenges of Streaming Application Development

Writing stream processors is at the heart of any streaming architecture. Yet, writing streaming applications that perform computations, aggregations, and filter streams of data takes specialized knowledge, specific access and is extremely time-consuming. This challenge creates a bottleneck in the organization and slows the development and usage of streaming systems.

- Streaming applications require more expert developers, which are hard to hire and train.
- Developers must understand lower-level languages like Java or Scala and be familiar with the powerful yet tricky streaming APIs.
- A few developers must support a vast array of production jobs and are often overloaded.
- Iterative development is almost impossible - and a long time to market the norm.
- Developers spend most of their time finding sources, gaining access, and wrangling data.
- Lack of a simplified method to interoperate with other streaming applications, machine learning models, and visualization tools.

## Simplify streaming application development with SQL Stream Builder

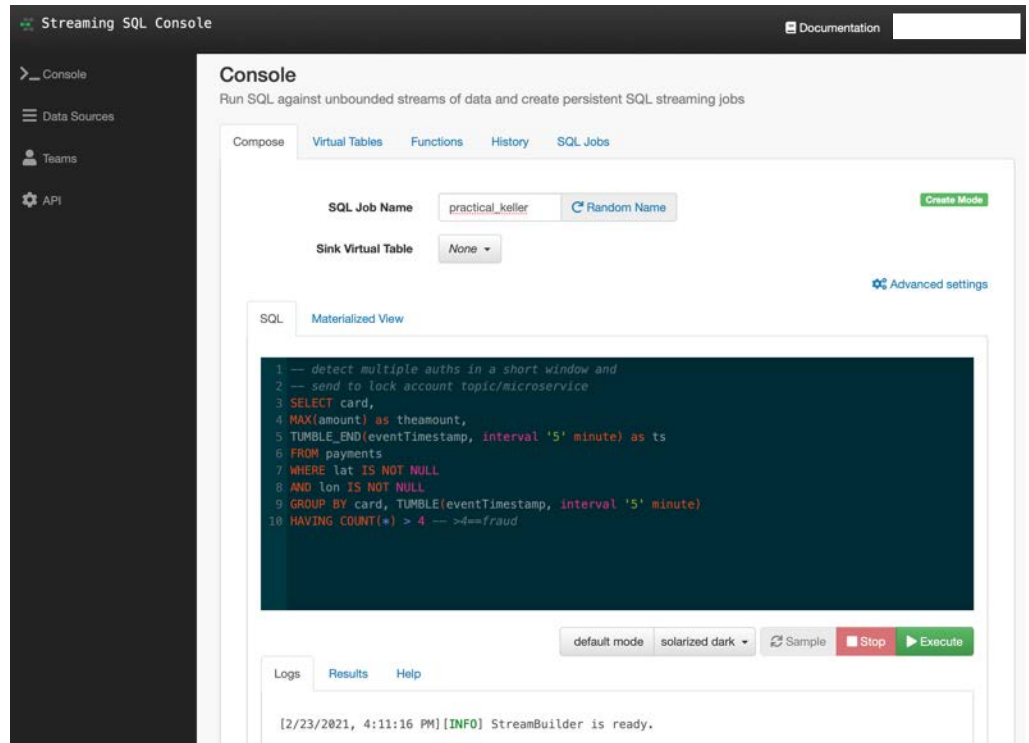
SQL Stream Builder allows developers, analysts, and data scientists to write streaming applications using industry-standard SQL. It provides an interactive experience - so the development process is quick, easy, and productive. It offers syntax checking, error reporting, schema detection, query creation, sampling results, and creating outputs with its powerful interface and APIs.

It provides an advanced materialized view engine to interface with applications, tooling, and services via REST. The underlying engine is designed for high performance, requires no maintenance, and queries can be of complex nature (reading nested JSON, for instance).

SQL Stream Builder continuously runs SQL via Apache Flink. Developers don't need to understand the Java programming language or complexities like watermarks. DevOps can manage, monitor, and scale jobs like any other Flink system. SQL jobs inherit and leverage the robust nature of Apache Flink. Jobs can be restarted and retain state (for fast restart and upgrade), have massive scalability, and a mature and robust run-time framework.

### About Cloudera

At Cloudera, we believe that data can make what is impossible today, possible tomorrow. We empower people to transform complex data into clear and actionable insights. Cloudera delivers an enterprise data cloud for any data, anywhere, from the Edge to AI. Powered by the relentless innovation of the open source community, Cloudera advances digital transformation for the world's largest enterprises. Learn more at [cloudera.com](https://cloudera.com)



### Key Features

- Streamlined user interface allows for interactive development experience - users can iterate on SQL statements, inspect data, and create processors. It has robust syntax checking, user-feedback, and logging.
- Automatic schema detection and source creation for JSON data sources. AVRO data sources use Schema Registry.
- Rich SQL processing grammar based on Apache Calcite. Event time, processing time, and custom time fields are allowed and integrated with Apache Kafka.
- Powerful input transformations in Javascript for handling foreign or messy feeds.
- Simple Javascript user functions interface that doesn't require restarts or recompilation.
- Materialized view engine that facilitates fast queries via any key over REST or PostgreSQL wire protocol.
- Rich integration with Apache Kafka for streaming sources and sinks including exactly-once processing.
- Join and enrich streams of data with batch data via SQL.