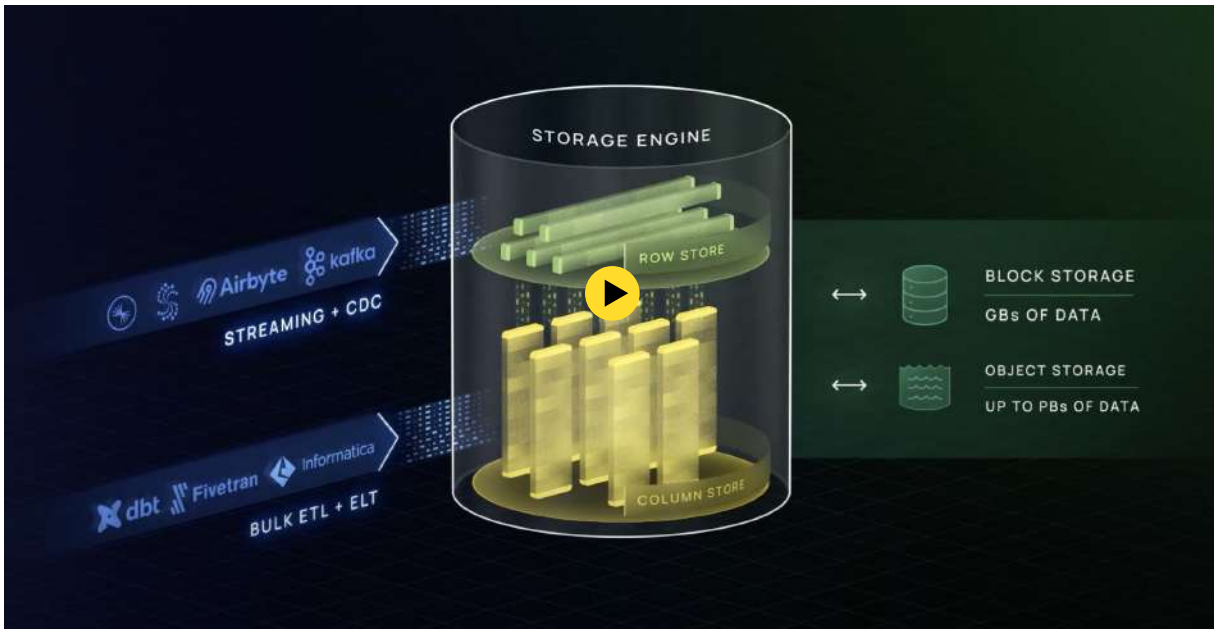
Yellowbrick

How Yellowbrick Enables Real-Time and Historical Data Analysis



Yellowbrick's patented architecture is designed for near real-time data loading, with a hybrid storage engine, PostgreSQL compatibility, and native and third-party connectors that enable instant query access from various streaming sources.

Don't Take Our Word For It: See How Our Customers Benefit

insightsoftware

Angles Enterprise for SAP simplifies access to supply chain data from SAP and delivers actionable business insights.

By switching to Yellowbrick from their existing data platform on AWS, they increased their streaming data load throughput by 300X and query performance on massive datasets by 4.5X, dramatically improving their user experience. The move saved them $40K per new customer onboarded, as their previous platform required costly per-customer instances for data isolation. Yellowbrick's Postgres front-end made data migration effortless, seamlessly integrating with their AWS data stack.

Learn More →

PROOF POINTS

Purpose Built for Real-time

Operational and event data arrives in real-time. Do you have the confidence to immediately present that data to end-users? Do you have different data solutions for real-time operational analytics and historical analytics? With Yellowbrick you can stream load or trickle-feed data, bulk load, and deliver high performance analytics in ONE unified platform with data instantly available to query. No architectural or DataOps gymnastics required.

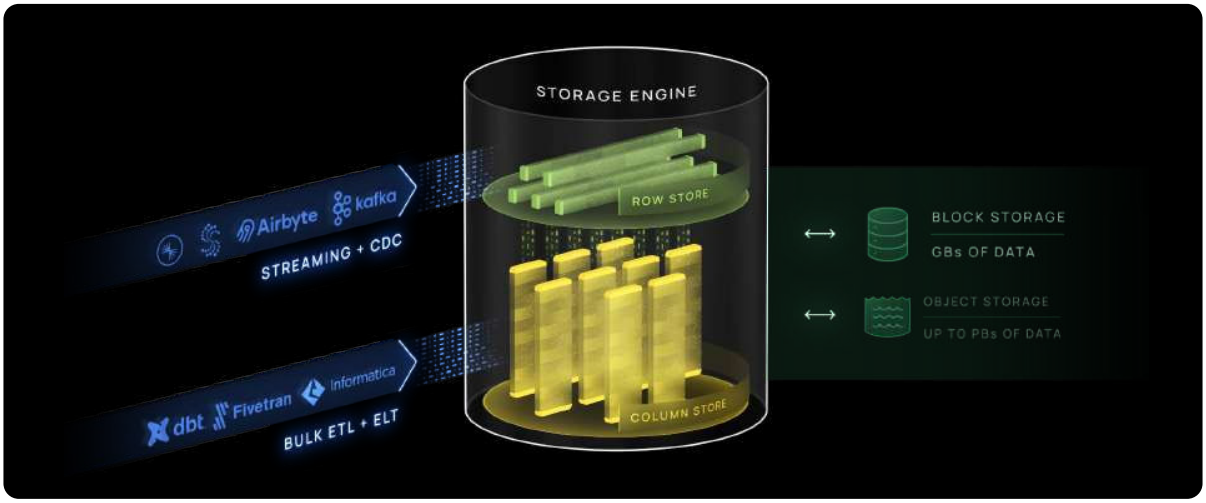
Dynamic decision-making requires analytics based on streaming data. Traditional databases optimized for batch processing struggle with frequent, small insertions typical of streaming data from sources such as Kafka, CDC tooling, and AWS Kinesis, which can delay data availability and hinder business decisions.

Innovative Architecture

Yellowbrick addresses this challenge with an architecture designed for loading data from near real-time streams. Yellowbrick's hybrid storage engine, its PostgreSQL compatibility, and its native and third-party data connectors to a wide variety of streaming sources ensure that data can be loaded and available to query almost instantly.

Hybrid Storage Engine

Most modern data warehouse implementations are backed by column stores alone. This approach can result in high data compression and good query performance but comes at the expense of single record and micro-batch load performance. Yellowbrick features a hybrid storage engine design that combines a front-end row store and a back-end column store that allows data to be inserted, bulk loaded and queried simultaneously without impacting data latency.



The data in a Yellowbrick table spans both the row store and column store and, from the perspective of a query or data load, appears as a single logical table. Data can be inserted into the row store on a record-by-record basis at high speed and is instantly accessible. Rows are automatically flushed into the column store over time. Bulk loads of large amounts of data are inserted directly into the column store via parallel connections to the workers, bypassing the row store. Transactions are preserved across the row and column store by using a common transaction log with a "read committed" level of isolation and multi-version concurrency control. With Yellowbrick, all of this happens transparently under the covers without any developer or DBA intervention.

Third Party and Native Streaming

Yellowbrick is PostgreSQL compatible, which means that almost any data connector that works with Postgres will work with Yellowbrick, enabling integration with a wide range of streaming platforms. Yellowbrick also includes native connectors to Kafka and Spark for streaming. There is third-party connectivity to a variety of other sources including AirByte, AWS Data Migration Service, Oracle GoldenGate, and Fivetran for change data capture use cases, and it also uses the standard PostgreSQL ODBC, JDBC and ADO drivers to support data streaming through SQL INSERT and PostgreSQL \COPY commands.