striim

5 Reasons Organizations Need Change Data Capture Solutions

and How to Choose the Right One

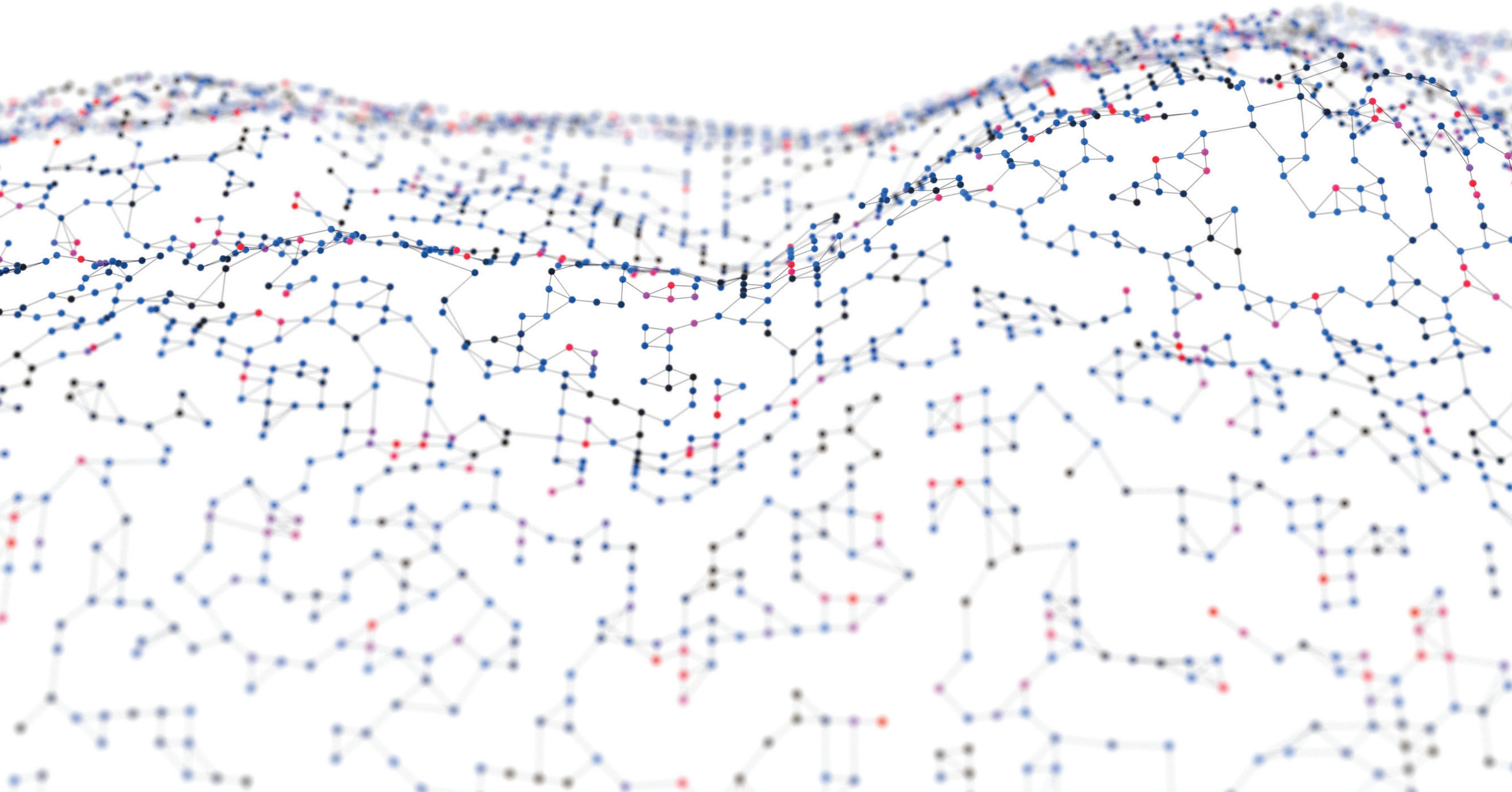


Table of Contents

CHAPTER 1:

An Introduction to Change Data Capture

> "It's a huge competitive advantage to see in real time what's happening with your data."
>
> *Hilary Mason,*
> *Data Scientist and Co-founder at Hidden Door (**Source**)*

Today, businesses can no longer afford to wait to analyze data from their data warehouse at the end of a week or month. The adoption of real-time data architectures is climbing at a rapid rate, making it essential to process and query data in real time or near real time. This allows companies to collect instant insights that govern improved business decision-making.

Traditionally, data warehouses stored historical data from operational systems for weekly or monthly analysis. They didn't store up-to-date data. Over time, a real-time data warehouse was needed because it became more crucial for businesses to deliver trusted and accurate information to the right consumers at the right time.

A real-time data warehouse collects data constantly, data that is queryable immediately after it arrives without performing time-consuming actions like compaction, aggregation, and processing. Similarly, queries also run faster in a real-time data warehouse. This speedy functioning of a real-time data warehouse makes it easy to perform real-time analytics.

However, the necessity for faster time-to-insight presents another dilemma. It needs the data to be captured from transactional systems quickly. Maintaining real-time data from when it's captured and making it available to decision-makers is technically difficult. You can't just copy your whole database for every instance to perform data analysis. Such replication can be resource-intensive for every query, causing delays.

When you have to process data instantly, it's more logical and efficient to copy new or modified data to your data warehouse via change data capture.

What Is Change Data Capture, and How Does it Work?

Change data capture (CDC) is a process that detects and captures changes in the source system (data source). These changes are recorded from the source system's tables in a specified period (usually minutes or seconds), and these changes are made available to a target system in an easily consumed relational format. The process also captures column information and metadata needed to make these changes to the target system's change table. A change table is a special type of database table that stores the change data from the source system's corresponding table. The process also stores the system metadata to maintain the change table.

A source system is a production database that stores the source tables. CDC captures changes from this source table. A source system can be an operational system, application, or a mainframe system. Recently, these source systems have also begun storing data generated by social media message streams and IoT sensors.

A target system is a repository to which CDC delivers the changes. A target system can be a data warehouse, data lake, or a cloud platform (e.g., AWS). Sometimes, message streaming platforms serve as an intermediary for transmitting data to different big data target systems.

The changes identified by CDC fall into four categories.

- **Inserts:** These are used to add one or more rows to a database.
- **Updates:** They are used to modify one or more fields in one or more rows.
- **Deletes:** They are used to delete one or more rows.
- **DDL:** These are changes made to the structure of a database via its data definition language (DDL). For example, they can be used to remove data types, columns, tables, and other database objects.

Methods to Implement Change Data Capture

You can implement CDC with a number of techniques.

Timestamp-based CDC

A simple approach for implementing CDC is to use a timestamp column in your database's table. This CDC approach uses a timestamp field in the source table to detect and collect changes in datasets. Sometimes, two timestamp fields are used for source systems; the first for storing the time at which a record was generated (CREATED_AT) and the second for storing the time at which the row was last modified (UPDATED_AT).

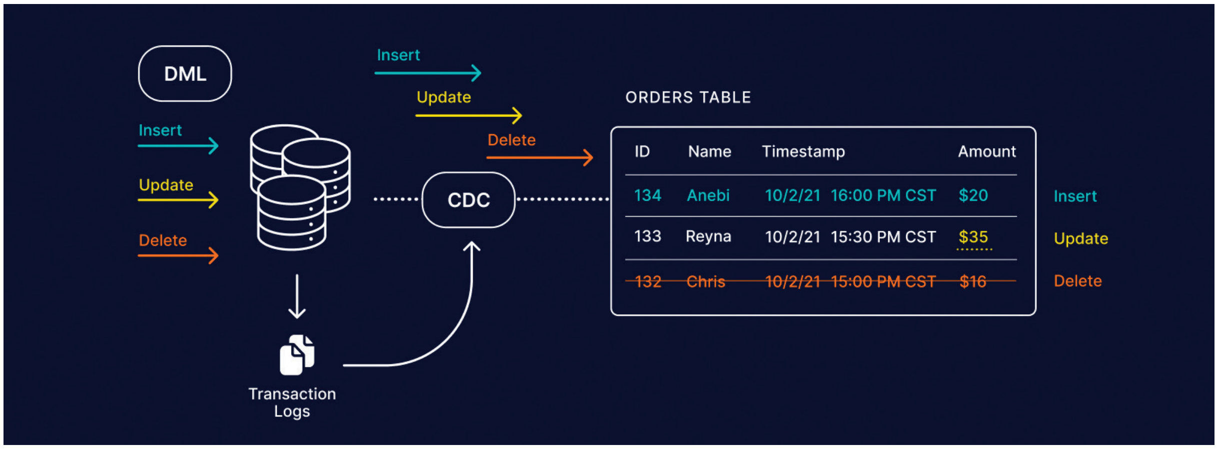
Implementation-wise, it's easier to use timestamps. If you have a basic system and you need to track updates to one of your core business units, such as sales, then timestamp-based CDC is your best option. However, there are some drawbacks to this approach as it can only fetch records that were updated after the last retrieval. If you delete a record from the table, timestamped-based CDC cannot capture the deletion operation. To track delete operations for the target system, you will have to write an additional script. In addition, this approach makes your CPU resources scan tables in the database and look for changed data. This adds overhead to the database.

Trigger-based CDC

Trigger functions are built-in functions in many modern database systems. They help to perform user-defined actions after specific events occur. For example, you can set a trigger after data is inserted in a table. Similarly, you can use triggers to record all changes and monitor them in a separate table known as a shadow table. Shadow tables can be used to store the entire row after a column is updated. They can also store the type of operation (insert, update, or delete) or primary key.

The disadvantage of this approach is that running triggers on operational tables can affect the database performance. Due to this, database administrators sometimes might disallow the use of triggers.

Log-based CDC



ID	Name	Timestamp	Amount
134	Anebi	10/2/21 16:00 PM CST	$20
133	Reyna	10/2/21 15:30 PM CST	$35
~~132~~	~~Chris~~	~~10/2/21 15:00 PM CST~~	~~$16~~

Log-based CDC collects recent database transactions from the source database's transaction logs

Transactional databases recover from crashes by storing all their changes in a transaction log. Log-based CDC collects recent database transactions from the source database's transaction logs. What makes log-based CDC more efficient than other approaches is that it captures changes without making any application-level changes. Unlike other approaches, it doesn't perform any scan on operational tables.

CHAPTER 2:

Why Do Organizations Need Change Data Capture?

Introducing CDC to your organization can address a number of technical challenges.

CDC Saves Computational Cost

With other approaches like table differencing, you have to run diffs on large tables to identify differences. You can use change-value selection to compare records for any columns (e.g., LAST_UPDATED). These approaches might capture changes like CDC, but they have a drawback in the form of higher computational cost. When you work with large databases, comparing tables via MINUS queries can take a great deal of time. Similarly, performing analytics queries on master databases can affect the speed of your application.

CDC, particularly log-based CDC, doesn't put load on the CPU of the production database. CDC doesn't require the use of additional SQL, which reduces the load on the system. Moreover, increment loading helps to minimize the impact on performance. Therefore, CDC can help you to save computational costs, even if your datasets continue to grow at an exponential rate.

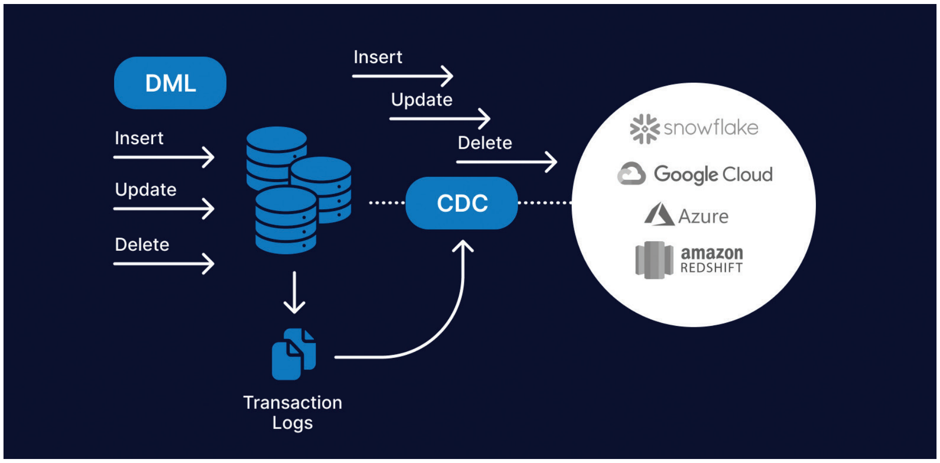
CDC Connects Multiple Database Systems

Teams that work with incompatible databases can use CDC to share data in real time. This is done by populating a replicate or analytics database, which collects data from all database systems. Anyone can access this staging database for their needs without adding load on the working of the main databases. This can be particularly beneficial for medium- and large-scale organizations that typically use multiple database systems.

Customer support roles frequently require information on products or services to get insights into customers. Sales teams often have access to this information, making it hard for customer support teams to look into relevant information quickly. You can use CDC to move any specific type of data to a data mart — a condensed version of a data warehouse — for customer support representatives. This way, they can instantly access the information they are looking for.

CDC can also come in handy for organizations that have just gone through a merger or acquisition. In these situations, it's common to deal with corporate systems that aren't compatible with each other. With CDC, you can combine data from these systems and load them into a single storage medium like a data warehouse or data lake. This way, you can empower your teams to perform analysis and reporting work with ease. You can also use these repositories to load data into future systems.

CDC Keeps the Data Warehouse Fresh



CDC helps organizations update their data warehouse in real time

CDC can help to move information to a data warehouse in an effective manner. With CDC, changes can be delivered to extract, transform, and load (ETL) tools in real time, making the process more efficient.

Usually, an ETL tool fetches data from the source system, performs relevant transformations on it, cleanses it, and then delivers the output to the data warehouse. All of these actions add up to a batch window — the time period during which the operational system moves data and is unable to take part in any mission-critical or operational function. Since many companies were satisfied with only updating their data warehouse daily or weekly, this wasn't much of an issue.

However, today it's more necessary for organizations to collect, process, and analyze real-time information. For that purpose, they have to update their data warehouse in real time, which can be possible through CDC.

CDC Improves Cache Consistency

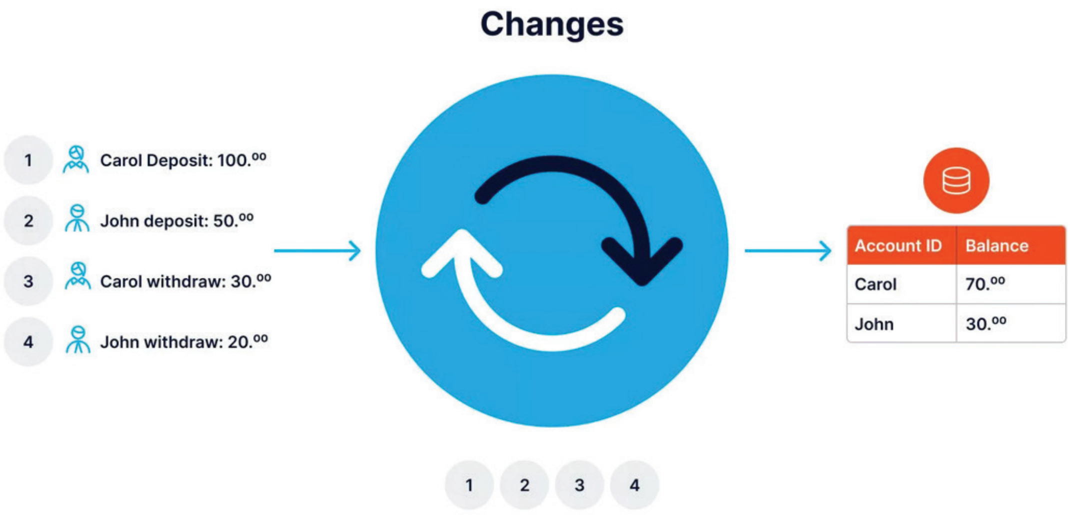
Suppose your application retrieves and stores information in a database, and you are looking for ways to improve response times. A common approach to improve access to data is to use an in-memory cache. But when other applications make updates to the database, it can cause a cache inconsistency problem. As a result, you might see outdated data in your application.

You can use CDC to synchronize the cache with your database in real time. A database reader can fetch data from the database, and a database writer can map the data from your database tables into the cache. This ensures that your cache will always have up-to-date data, allowing your application to work reliably.

CDC Simplifies Database Migration and Replication

Whenever organizations migrate data from legacy systems, they are prone to challenges during the migration of data in the application tier and database. While the migration is in process, it can also be tough to maintain user access to applications without any interruption.

CDC can be useful because it supports zero-downtime cloud migration. You can use CDC to load data from the source database and continuously capture all changes that occur in the loading process. Once the initial load finishes, you can apply the changes to the target system's table. This real-time approach can maintain consistency between your legacy and cloud databases. Here's a diagram that shows how a financial company's target database will look after changes are captured and delivered to the target system's table.


Changes
1 Carol Deposit: 100.^{00}
2 John deposit: 50.^{00}
3 Carol withdraw: 30.^{00}
4 John withdraw: 20.^{00}
1 2 3 4

Account ID	Balance
Carol	70.^{00}
John	30.^{00}

CHAPTER 3:

How to Choose the Right Change Data Capture Tool

One of the ways to implement CDC is to build an in-house solution of your own. However, this approach comes with a host of drawbacks.

1. In-house developers are always busy with a backlog of requests. By moving their attention to CDC, you might distract their focus from key revenue-generating projects.
2. Building the CDC solution is a job half done; you also have to consider the maintenance cost of your custom solution, which includes factoring in changes in database schemas and logs.
3. CDC implementation is complicated by a number of technical challenges. These include access to logs, varying log formats, and tackling different database vendors.

A better approach is to invest in a commercial CDC solution that's designed to handle enterprise workloads. There are several key features to look for when adopting a third-party CDC tool.

Non-Intrusive CDC

While implementing CDC, you need to assess the load it brings to the source system. This impact can vary. Choose a solution that inflicts the minimum load on your source system, especially when it comes to cost, operations, and maintenance.

A CDC solution that's less invasive will add timestamp fields to make changes to the source tables' schema. While updating timestamps, there's some impact on the source system's processing and storage requirements. Similarly, a CDC solution can also use trigger-based CDC. Although these approaches don't affect the application that's making the changes, it does increase processing time within the source system. This process can hog some resources that are used by the operational system.

If you are looking for the least invasive approach with minimal impact on the source system, then you need a CDC tool that supports log-based CDC and doesn't access the source system. Instead, it only accesses the log of the source system. This type of CDC can be handled independently, preventing it from sharing resources with the operational system.

Batch and Real-Time Delivery

When it comes to CDC, not all applications have the same latency needs. A reliable CDC solution should have the ability to adapt by offering both batch and real-time delivery.

For instance, suppose you have to use CDC for a data warehouse that's updated every six hours. You can use ETL tools to perform this update. In order to process six hours' worth of records, it's more logical to read all changes and process them in a batch.

However, if you have two applications that depend on real-time data and must be synchronized in a timely fashion, you need a CDC solution that can offer real-time capabilities.

Change Filtering

You need to look for a CDC solution that enhances efficiency by minimizing the amount of data that has to be processed. This is done through support for filters that let you reduce the amount of information at hand and ensure that only the applicable records get delivered.

You can use filters for the following purposes:

1. Customize the delivery of records based on the form of change (e.g., inserts, deletes)
2. Enable the delivery of records based on changes in a specific field
3. Choose a subset of fields from the original record that has to be processed

Filters are beneficial for the source system as they cut the number of records that have to be traversed.

Management of Non-Relational Data

Often, legacy systems store critical information in a non-relational structure, which can be tough to handle. Organizations that have this type of infrastructure should look for a CDC solution that can capture data from non-relational data sources smoothly.

If you are going to process the data changes with an enterprise application integration (EAI) tool — usually in XML — see if your solution offers mapping functionality to map your data source into an XML document, along with a corresponding XML schema to show the hierarchy of the original record.

If you are going to process the information with an SQL-based ETL tool, find a solution that offers normalization capability for your non-relational data and shows a metadata model in a relational structure.

Get Striim to Meet Your Enterprise CDC Needs

If you are looking for a CDC solution that can manage enterprise workloads with ease, then look no further than Striim. Here's what Striim offers:

1. Log-based CDC

Striim comes with support for log-based CDC, the most efficient CDC technique. Striim relies on log-based CDC to collect data from MySQL, Oracle, SQLServer, MongoDB, and other popular database systems. As a result, you can reduce the CPU overhead on your source system and put an end to making application changes.

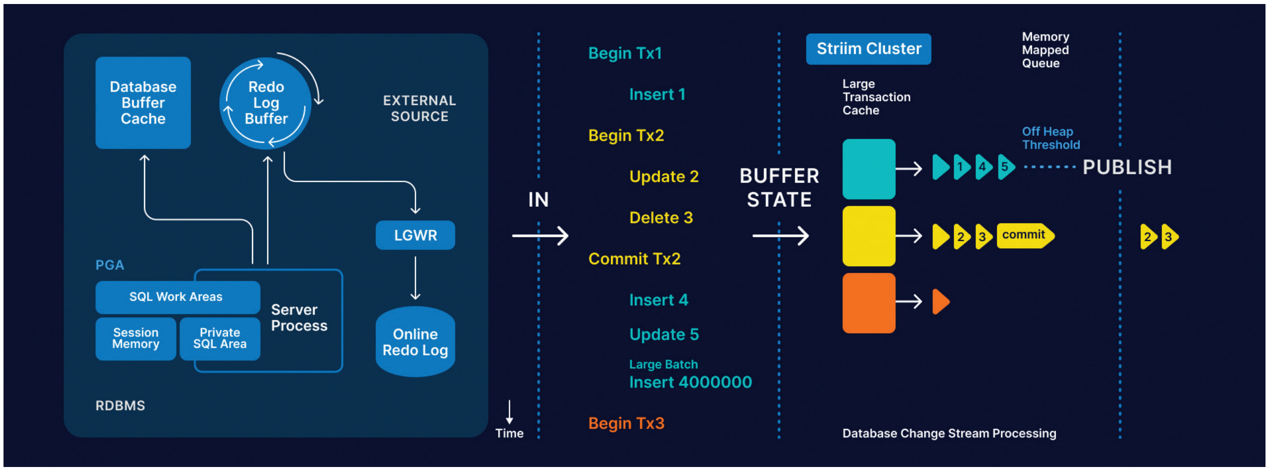
2. Data ingestion

Striim ingests data from multiple sources concurrently, combining database transactions with unstructured and semi-unstructured data. Striim lets you combine real-time transactional data from online transactional processing (OLTP) systems (e.g., online airline ticket booking) with the following data in real time:

- Messaging systems' events
- NoSQL data
- Sensor data
- Real-time log data
- Cloud application data

This data merging is useful for generating more in-depth and credible information for your organization.

3. Support for long-running transactions



Striim users can configure an off-heap threshold to buffer large transactions to disk.

Striim prevents data loss by providing support for long-running transactions. If you work with large workloads, you can set an off-heap threshold to buffer your long-running transactions smoothly to disk, while at the same time limiting the performance overhead to a minimum.

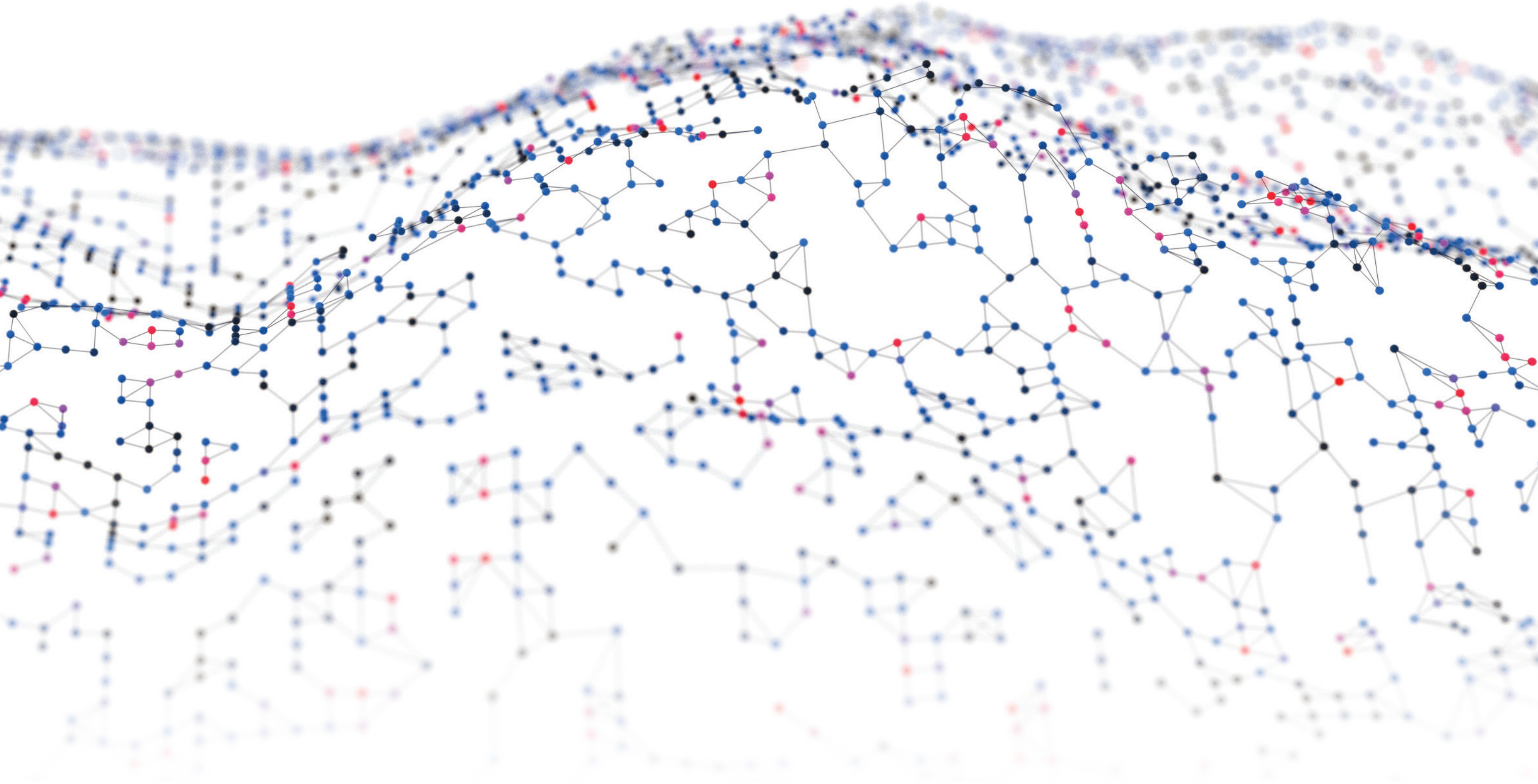
4. Turning change data into time-sensitive insights

Thanks to Striim's customizable time windows, you can perform streaming analytics, enabling you to get instant insights from your data in motion. For instance, suppose you are looking to move financial transactions with Striim. You can create real-time dashboards in Striim to notify your consumers about potential fraud incidents before Striim sends your data to your analytics tool.

5. End-to-end data integration

Striim isn't merely built to ingest change data. If you are looking to perform other actions, such as processing, securing, scaling, monitoring, and delivering, Striim can help you do everything in an effective way. This is possible due to the following features:

- **Built-in checkpoint for reliability:** During data movement and processing, Striim records and tracks all operations. If an outage occurs, it can simply look up the transaction from where it was left off. This prevents data from getting missed or duplicated.
- **Transactional integrity:** Striim maintains transactional context while moving committed transactions after ingestion of change data. This context is saved during other data operations — movement, processing, and delivery — allowing users to generate replica databases.
- **In-flight change data processing:** Striim provides data transformers. Transformers are functions and packaged actions that help to modify data. They are especially useful for converting raw data into a presentable format. Striim also comes with in-memory stream processing capabilities for filtering, aggregating, masking, transforming, and enriching change data after it's captured. This allows you to transform your change data into a format that can be easily consumed by your target audience.
- **Distributed processing in a clustered environment:** Striim can improve scalability and availability with its clustered environment. This way, you don't have to depend on using third-party products for building clusters. You can use affordable hardware to scale out for larger data volumes.
- **Data delivery validation:** Striim performs comparisons between source and target environments continuously while data is in motion. This is done to maintain consistency between databases and ensure that all the captured changes have been delivered to the target system. This can be extremely useful for data migration, where avoiding data loss is a priority.
- **Pre-packaged applications for initial load and CDC:** Striim provides **example integration applications** that contain initial load and CDC for PostgreSQL environments. You can use these integration applications to set up data pipelines quickly, as well as use them as a blueprint for working with other CDC sources.



About Striim

Striim was founded with a simple goal of helping companies make data useful the instant it's born.

Striim's unified, real-time data streaming and integration platform for analytics and operations collects data in real time from enterprise databases (using non-intrusive change data capture), log files, messaging systems, and sensors, and delivers it to virtually any target on-premises or in the cloud with sub-second latency enabling real-time operations and analytics.

Try it now at **go2.striim.com/free-trial**

Contact us at:

Tel: +1 650 241 0680
Email: **sales@striim.com**
Web: **www.striim.com**

striim