<https://www.integrate.io/blog/how-to-implement-change-data-capture-in-sql-server/#two>

**How to Implement Change Data Capture in SQL Server**

 Apr 11, 2023

Every organization wants to stay on the cutting edge of technology, making smart and data-driven decisions. Microsoft SQL Server data collection will help capture valuable information and save it in a relational database or data warehouse for analysis.

However, ensuring that company information and data integration remains fresh and relevant can be a very time-consuming process. That’s where CDC can make all the difference.

[Change data capture or CDC](https://www.integrate.io/blog/what-is-change-data-capture/) ensures that enterprise data is always up-to-date and can make companies’ data workflows dramatically faster and more efficient. So what is change data capture, exactly, and what are the different types of CDC? Also, how is CDC implemented in SQL Server, and how does it integrate with ETL? Read on to discover the answers to these questions.

**Here are 5 key takeaways from this article:**

* Change Data Capture (CDC) is a feature in SQL Server that allows you to capture insert, update, and delete operations performed on a SQL Server table and write them to a separate table. This can be useful for a variety of purposes, including auditing, replication, and data warehousing.
* To enable CDC on a SQL Server table, you need to first enable it at the database level and then enable it on the specific table. Once enabled, SQL Server will automatically create a separate table to store the changes.
* CDC uses a special type of table called a change table to store the captured changes. The change table has a similar structure to the original table but includes additional columns to track the type of change, the time of the change, and the transaction that caused the change.
* You can use the CDC functions provided by SQL Server to query the change table and retrieve the captured changes. These functions allow you to filter the changes by time range, transaction ID, and other criteria.
* When implementing CDC, it's important to consider performance and scalability. CDC can generate a significant amount of additional workload on your SQL Server instance, so you may need to optimize your database design and hardware infrastructure to handle the increased load. Additionally, you should monitor your CDC jobs regularly to ensure they are running smoothly and not causing any issues.

Change Data Capture (CDC) is a powerful feature in SQL Server that enables you to track changes made to your database. In this article, we'll walk you through the process of implementing CDC in SQL Server, from enabling it at the database and table levels to working with change tables and CDC functions. We'll also cover important performance considerations and best practices for implementing CDC in your SQL Server environment. Whether you're new to CDC or looking to optimize your existing implementation, this article has something for you.

**Table of Contents**

* [What is Change Data Capture](javascript:;)?
* [Types of Change Data Capture](javascript:;)
* [Change Data Capture and ETL](javascript:;)
* [Implementing CDC in SQL Server](javascript:;)
* [CDC and SQL: How Integrate.io Can Help](javascript:;)

**The Unified Stack for Modern Data Teams**

Get a personalized platform demo & 30-minute Q&A session with a Solution Engineer

Top of Form

GET A LIVE DEMO

Bottom of Form

**What is Change Data Capture?**

Change data capture, or CDC, is a set of software processes and techniques that identify changes in source tables and source databases, and then transfer those database changes. When companies use CDC, they can usually detect changes in real-time. In most cases, the changed entries move through data replication to a specific target location; companies can then use this updated data for business intelligence (BI) and data analytics workflows.

Time is money when it comes to business. Change data capture is an ideal solution for companies looking to work with data more efficiently, as CDC works in real-time movement. After data collection in SQL Server, CDC helps move this information to a data warehouse, data lake, or other databases in real-time or near-real-time. The data movement efficiency that change data capture provides is extremely beneficial for organizations.

***Related Reading:***[What is Change Data Capture?](https://www.integrate.io/blog/what-is-change-data-capture/)

**Types of Change Data Capture**

CDC has two main types. First, companies can perform change data capture by log-based CDC; second, they can use trigger-based CDC.

**Log-based CDC**

In[log-based CDC](https://www.integrate.io/), the change data capture solution examines a database’s[transaction log](https://docs.microsoft.com/en-us/sql/relational-databases/logs/the-transaction-log-sql-server?view=sql-server-ver15&utm_source=xp&utm_medium=blog&utm_campaign=content). During this process, the CDC solution reads the file to uncover the source system changes. This metadata information is stored in CDC change tables. Then, it executes data replication of these source changes to the target data store.

The **pros** of log-based CDC are:

* High reliability with no missed changes.
* Minimal impact on the production database system.
* No requirements to change the production database's schemas or the need to use additional SQL Server CDC tables.

The **cons** of log-based CDC are:

* Works only with databases that support log-based CDC.
* High complexity.

**Trigger-based CDC**

In trigger-based CDC, the change data capture solution uses database triggers. During this process, the CDC solution runs when another event occurs. These database triggers can decrease the overhead that results from extracting changes. However, they also add overhead to the source systems because they require a certain amount of run time each time the existing database refreshes.

The **pros** of trigger-based CDC are:

* Easy implementation.
* Changes can happen quickly.
* Shadow tables can provide a detailed log of all transactions.
* Receives direct support in the SQL API for some databases.

The **cons** of trigger-based CDC are:

* Can experience trigger overload.
* Triggers may be disabled during certain operations.
* Significantly reduces the overall performance of the database by requiring multiple writes to a database every time a user inserts, updates, or deletes a row.

**Change Data Capture and ETL**

The real perk of change data capture is that it can save companies from focusing on unnecessary data work. By implementing MS SQL CDC, the company ensures that its system only concentrates on new updates to records, rather than an entire SQL Server database.

The benefits of change data capture can also prove useful for ETL. The most commonly used and dominant type of data integration is ETL or Extract, Transform, Load. In ETL, the process extracts information from one or more data sources, cleans it, and transforms it as necessary. Finally, the process delivers that information to a data warehouse, data lake, or other database types.

***Related Reading:***[What is ETL?](https://www.integrate.io/blog/what-is-etl/)

Pairing SQL change data capture with ETL has the potential to save companies a great deal of time and effort, compared to running a traditional ETL system. This is because traditional ETL has a slow transformation step. Change data capture SQL Server solutions help improve the time required to carry out the data transfer and decrease the resources required for the entire ETL process.

***Related Reading:***[ETL vs ELT: 5 Critical Differences](https://www.integrate.io/blog/etl-vs-elt/)

**Implementing CDC in SQL Server**

Now that we've discussed the types of change data capture and how they relate to ETL, it’s time to look at how to implement CDC in SQL Server. Change data capture records, inserts, updates, and deletes activity that applies to an SQL Server table, which means organizations can capture changes in SQL Server data by using the SQL Server change data capture feature.

However, the data system must meet certain prerequisites before you can enable CDC SQL Server. These prerequisites include:

* Having “sysadmin” privileges
* Running SQL Server Developer, Enterprise, or Standard Edition, as the web does not support CDC functionality
* Ensuring the SQL Server Agent runs on an SQL Server instance

Once the company system meets these prerequisites, the user can use the following steps to implement SQL Server change data capture.

1. Open the SQL Server Management Studio and create a database.
2. Create a table.
3. Enable CDC on the database.
4. Define the specific table on which to enable change data capture.
5. Insert the values into the table.
6. Verify that the change data capture is working.

What does SQL CDC look like under the hood in terms of technical implementation? The answer can get a little messy and technically involved:

* **dbo.cdc\_jobs** is a Microsoft SQL Server system table that keeps track of the CDC parameters for capture jobs and cleanup jobs. This includes settings such as the database ID, the type of job to run, the number of seconds between polling for new data, and much more.
* **sys.sp\_cdc\_enable\_db** is a stored procedure for enabling CDC on the given database, while sys.**sp\_cdc\_enable\_table** turns on CDC for a given table. To run these procedures, type “exec sys.sp\_cdc\_enable\_db” and “exec sys.sp\_cdc\_enable\_table”, respectively. These commands take multiple arguments: for example, *source\_name* is the table name of the source on which you are enabling CDC, and *role\_name* is the name of the role that controls access to change data.
* Implementing CDC in SQL Server will also require familiarity with DDL (Data Definition Language) and DML (Data Manipulation Language) SQL commands.

Ultimately, by implementing MSSQL CDC, organizations can reduce the time spent on data integration tasks and ensure the system handles information more efficiently, changing data in real-time.

**CDC and SQL: How Integrate.io Can Help**

Is your company in need of a simple yet efficient solution for ETL and SQL Server CDC? We can help.[Integrate.io](https://www.integrate.io/) is a powerful and feature-rich yet user-friendly cloud-based ETL (Extract, Transform, Load) solution. The Integrate.io platform's drag-and-drop interface makes it easy to build data pipelines for automated data flow across various sources and destinations.

Thanks to Integrate.io’s code-free data integration, everyone from data professionals to non-technical team members can build rich data pipelines. Whether you’re a developer or non-developer, your company can benefit from using CDC and ETL solutions.

**The Unified Stack for Modern Data Teams**

Get a personalized platform demo & 30-minute Q&A session with a Solution Engineer

Top of Form

GET A LIVE DEMO

Bottom of Form

 Tags: [big data,](https://www.integrate.io/blog/tag/big-data/)[CDC,](https://www.integrate.io/blog/tag/cdc/)[SQL Server](https://www.integrate.io/blog/tag/sql-server/)

**Share This Blog Post**

**Related Readings**

**[Mastering SQL Queries in Excel](https://www.integrate.io/blog/mastering-sql-queries-in-excel/)**

[July 24, 2024](https://www.integrate.io/blog/mastering-sql-queries-in-excel/)

[by Mark Smallcombe](https://www.integrate.io/blog/mastering-sql-queries-in-excel/)

[Mastering SQL queries in Excel is game-changing for anyone who works with large datasets and complex data management tasks. Our latest post offers a comprehensive guide to effectively using SQL...](https://www.integrate.io/blog/mastering-sql-queries-in-excel/)

**[ETL Developer vs Data Engineer: Key Differences](https://www.integrate.io/blog/etl-developer-vs-data-engineer/)**

[June 29, 2024](https://www.integrate.io/blog/etl-developer-vs-data-engineer/)

[by Donal Tobin](https://www.integrate.io/blog/etl-developer-vs-data-engineer/)

[Data management is one of today's most important business functions. Organizations can't use data effectively without a solid grasp of what data management entails. Businesses look to ETL developers and...](https://www.integrate.io/blog/etl-developer-vs-data-engineer/)

**[How Enterprise Automation Transforms Workflows](https://www.integrate.io/blog/how-enterprise-automation-transforms-workflows/)**

[June 13, 2024](https://www.integrate.io/blog/how-enterprise-automation-transforms-workflows/)

[by Donal Tobin](https://www.integrate.io/blog/how-enterprise-automation-transforms-workflows/)

[Are you noticing bottlenecks or disadvantageous data silos in your enterprise workflows? Enterprise automation uses software programs and other technologies to perform repetitious business processes, and it could be the...](https://www.integrate.io/blog/how-enterprise-automation-transforms-workflows/)

**Subscribe To  
The Stack Newsletter**

Top of Form

Bottom of Form

[hello@integrate.io](mailto:hello@integrate.io)[+1-888-884-6405](tel:+1%20888-884-6405)

[©2024 Integrate.io](https://www.integrate.io/blog/how-to-implement-change-data-capture-in-sql-server/)

SIGN UP FOR “THE STACK” - OUR MONTHLY NEWSLETTER

Top of Form

Bottom of Form

Solutions

* [Solutions Home](https://www.integrate.io/solutions/)
* [Connectors](https://www.integrate.io/integrations/)
* [Marketing](https://www.integrate.io/solutions/media-and-entertainment/)
* [Sales](https://www.integrate.io/solutions/business-intelligence/)
* [Support](https://www.integrate.io/solutions/customer-360/)
* [Developers](https://www.integrate.io/solutions/engineering-team/)
* [Release Notes](https://www.integrate.io/product/changelog/)

Support

* [Blog](https://www.integrate.io/blog/)
* [Live Chat](https://www.integrate.io/blog/how-to-implement-change-data-capture-in-sql-server/#contact)
* [Support & Resources](https://www.integrate.io/support/)
* [Developers](https://github.com/xplenty/xplenty-api-doc-v2)
* [Documentation](https://www.integrate.io/support/#documentation)
* [Documentation API](https://www.integrate.io/docs/api/)
* [Security](https://www.integrate.io/security/)
* [Service Status](https://status.xplenty.com/?__hstc=114807128.daf88c90f918f3be957aa76efbf7f286.1722910012351.1722910012351.1722910012351.1&__hssc=114807128.1.1722910012352&__hsfp=2112289063)
* [Privacy Policy](https://www.integrate.io/privacy/)
* [Terms of Service](https://www.integrate.io/tos/)
* [Glossary](https://www.integrate.io/glossary/)
* [Consent Preferences](https://www.integrate.io/blog/how-to-implement-change-data-capture-in-sql-server/)

Company

* [Customers](https://www.integrate.io/customers/)
* [White Papers](https://www.integrate.io/books-and-guides/)
* [Webinars](https://www.integrate.io/webinars/)
* [About](https://www.integrate.io/about/)
* [Partners](https://www.integrate.io/partners/)
* [Join Us](https://www.integrate.io/careers/)

Language

* [English](https://www.integrate.io/blog/)
* [日本語](https://www.integrate.io/jp/blog/)
* [한국어](https://www.integrate.io/ko/blog/)

**1**