DATA-CORE SYSTEMS

Modernize Real-Time Data Analytics

A Whitepaper on Change Data Capture (CDC)

Data-Core Systems Inc.

www.datacoresystems.com

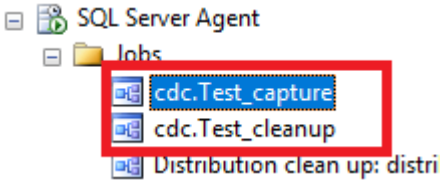
Introduction

Change Data Capture, also known as CDC, was first introduced in SQL Server 2008 version, as a helpful feature to track and capture the changes that are performed on the SQL Server database tables, with no additional programming efforts. Before SQL Server 2016, Change Data Capture could be enabled on a SQL Server database only under the SQL Server Enterprise edition, which is not required starting from SQL Server 2016.

Change Data Capture tracks the INSERT, UPDATE and DELETE operations on the database table, and records detailed information about these changes in a mirrored table, with the same columns structure of the source tables, and additional columns to record the description of these changes. SQL Server writes one record for each INSERT statement showing the inserted values, one record for each DELETE statement showing the deleted data and two records for each UPDATE statement, the first one showing the data before the change and the second one showing the data after performing the change.

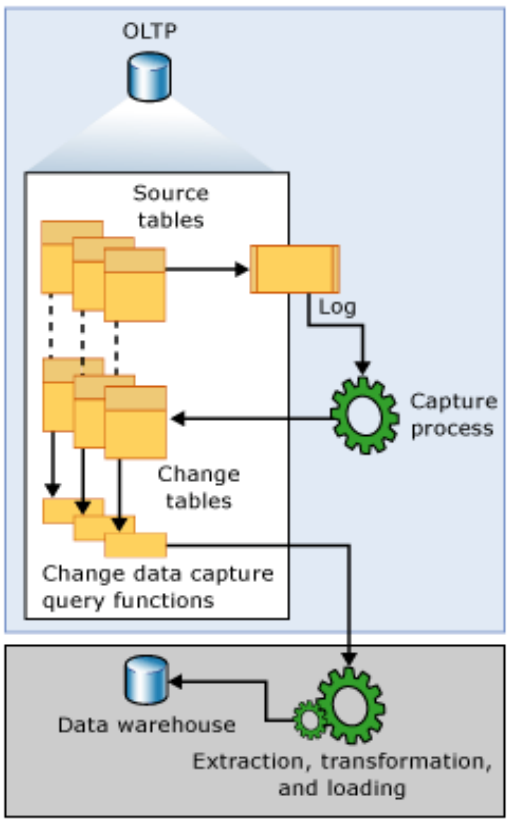
Architecture

Change Data Capture requires that a SQL Server Agent is running on a SQL Server instance. When the feature is enabled on a SQL Server database table, two SQL Server Agent jobs are created for that database. The first job is responsible for populating database change tables with the change information and the second job is responsible for cleaning up the change tables by deleting the records older than the configurable retention period of 3 days.

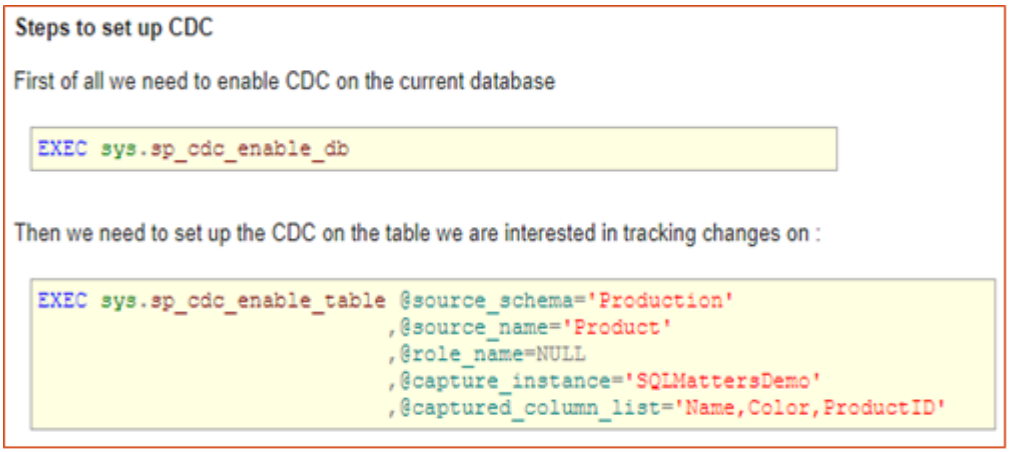


Change Data Capture depends on the SQL Server Transaction Log as the source of the data changes. When a change is performed, this change will be written to the Transaction Log file.

If the CDC feature is enabled on that table, the transaction log replication log reader agent, which acts as the capture process for CDC feature, will read the change logs from the transaction log file, add the metadata information about these changes and write it to the associated CDC change tables.



The source of change data for CDC is the SQL Server transaction log. As inserts, updates, and deletes are applied to tracked source tables, entries that describe those changes are added to the log. The log serves as input to the capture process. This reads the log and adds information about changes to the tracked table's associated change table. Functions are provided to enumerate the changes that appear in the change tables over a specified range, returning the information in the form of a filtered result set.



CDC tracks the INSERT, UPDATE and DELETE operations on the database table and records it in a mirrored table, with the same column structure of the source table, and additional columns to record the description of the changes.

The additional columns include:

- **__$start_lsn** and **__$end_lsn** that show the commit log sequence number (LSN) assigned by the SQL Server Engine to the recorded change
- **__$seqval** that shows the order of that change related to other changes in the same transaction
- **__$operation** that shows the operation type of the change, where 1 = delete, 2 = insert, 3 = update (before change), and 4 = update (after change)
- **__$update_mask** that is a bit mask defined for each captured column, identifying the updating columns

These additional columns make it easier to monitor the database changes for security or auditing purposes, or incrementally load these changes from the OLTP source to the target OLAP data warehouse, using T-SQL or ETL methods.

Metadata in CDC

Metadata is a set of data that gives information about other data. In the context of replication and CDC, primary categories and examples of metadata include infrastructure servers, sources, targets, processes, resources, users, usernames, roles and access controls, logical structure schemas, tables, versions, data profiles, data instances, and files and batches. Metadata plays a critical role in traditional and modern data architectures.

By describing datasets, metadata enables IT organizations to discover, structure, extract, load, transform, analyze, and secure the data itself. Replication processes, be they either batch load or CDC, must be able to reliably copy metadata between repositories.

METADATA$ACTION	METADATA$ISUPDATE	METADATA$ROW_ID
INSERT	TRUE	943f305c36accd512bc8d90700ea96d1bee777de
DELETE	TRUE	943f305c36accd512bc8d90700ea96d1bee777de

Advantages

- Can be configured to only track certain tables or columns.
- Able to handle model changes to a certain degree.
- Does not affect performance as heavily as triggers because it works with the transaction logs.
- Easily enabled/disabled and does not require additional columns on the table that should be tracked.

Benefits of Incremental Loading

Incremental loading is always a big challenge in data warehouse and ETL implementation. In the enterprise world we face billions of records in fact tables. It wouldn't be practical to load those records every night, as it would have many of the following downsides:

1. ETL process will slow down significantly, and can't be scheduled to run on small periods.
2. Performance of the source and destination server will be affected and downtime of these systems would be longer.
3. More resources will be required to maintain the process, such as better processors, more RAM, etc., and adding these won't help so much at the end, because the amount of data is increasing over time.

There are different methods to identify a change set and implement the ETL process in a way that only transfers the change set. Incremental loading is an efficient method especially when working with a source database that supports CDC technology.

The biggest benefit of log-based change data capture is the asynchronous nature of CDC. Changes are captured independent of the source application performing the changes. Dedication and smart software engineers can take care of the biggest challenges. Log-based CDC is generally considered the superior approach to change data capture that can be applied to all possible scenarios including systems with extremely high transaction volumes.

CDC is a technology that continuously scans source data systems for changes, identifies them, and delivers those changes to the data warehouse in real-time, so business intelligence applications access only the most recent data.

Change Data Capture technology entails three major benefits:

- Changes in the source systems are delivered to the data warehouse in real-time.
- Transactional databases are unaffected because production doesn't need to be paused.
- By reading the database's log, CDC gets the complete list of all data changes in their exact order of application, so there's no chance of missing data.

Considering CDC, triggers are not reliable because they may be disabled when certain operations take place. Users may also turn off triggers. Triggers capture changes only to the data, not to the table definition. Triggers will have an impact on the DB performance, due to resources sharing, locks, etc. The CDC will leverage the changes applied to the logs as opposed to the actual database, as a result it will have "no" impact on the DB performance, so it is "better".

In conclusion, we can deduce that incremental loading is more efficient than full reloading unless the operational data sources happen to change dramatically. Thus, incremental loading is generally preferable.

However, the development of ETL jobs for incremental loading is ill-supported by existing ETL tools. In fact, currently separate ETL jobs for initial loading and incremental loading have to be created by ETL programmers. Since incremental load jobs are considerably more complex, their development is more costly and error-prone.

To overcome this obstacle in this scenario we proposed the Change Data Capture (CDC) technique.

Limitations

Change Data Capture can be easily used to audit only the database DML changes and no option to monitor the SELECT statement, with the negligible configuration effort. On the other hand, to consider CDC as a SQL Server Audit solution, it requires significant maintenance and administration effort. This includes automating an archiving mechanism, as the tracking data will be kept in the change table for a configurable number of days and will be stored in the same or different data file that should be also monitored and maintained.

In addition, the change tables will be stored under each database, and a function will be created for each tracked table. This makes it cumbersome and requires significant programming effort to create a consolidated auditing report that reads the DML changes information from all tables under the same database, from all databases under the same instance, or cross multiple instance.

Another limitation for the CDC feature as a SQL Server Audit solution is the difficult process that is required to handle the DDL changes on CDC enabled table, as having the Change Data Capture enabled on the source table will not prevent performing DDL changes on that table.

Conclusion

In summary, the CDC helps modernize data environments by enabling faster and more accurate decisions, minimizing disruptions to production, and reducing cloud migration costs. An increasing number of organizations are turning to CDC, both as a foundation of replication platforms and as a feature of broader extract, transform, and load (ETL) offerings such as Microsoft SQL Server Integration Services (SSIS). It uses CDC to meet the modern data architectural requirements of real-time data transfer, efficiency, scalability, and zero-production impact.

The developers have tried to create systems that record all the changes made to the data in a database application. At last, with SQL Server 2008, have a robust way, CDC, that comes 'out of the box' to deliver this functionality in a standard way. This should be useful for auditing databases and for tracking obscure problems that require to know exactly when and where a change to a base table was made.

Data is at the core of what we do.

Our world is being re-imagined through Analytics, Artificial Intelligence and Automation. Data-Core Systems is a digital transformation solution provider helping businesses reshape their future. We are a proven partner with a passion for client satisfaction, combining technology innovation, business process expertise and a global, collaborative workforce.

DATA-CORE SYSTEMS

Data-Core Systems Inc.

1500 John F. Kennedy Blvd.
Suite 624
Philadelphia, PA 19102

Tel: 215 243 1990

Toll Free: 877 327 4838 Fax: 215

243 1978