عنوان مقاله: مقایسه Temporal Tables با Change Data Capture و Change Tracking در SQL Server | قسمت دوم

نویسنده مقاله: تیم فنی نیک آموز

تاریخ انتشار: فروردین ۱۴۰۱

منبع: https://nikamooz.com/sql-server-temporal-tables-vs-change-data-capture-vs-change-tracking-part۰۲

مساله

در قسمت اول، ما در مورد نحوه عملکرد ویژگی Change Tracking در SQL Server صحبت کردیم. Change Tracking فقط آخرین تغییر ایجاد شده در ردیف را ذخیره می‌کند. تاریخچه تغییرات قبلی را حفظ نمی‌کند. اگر چه کاربرد محدودی دارد، برخی از برنامه‌ها ممکن است فقط به این قابلیت تازه‌سازی ساده نیاز داشته باشند و برای اهداف ردیابی داده‌ها به temporal table یا CDC نیاز ندارند.

در این مقاله می‌خواهیم ببینیم که یکی دیگر از ویژگی‌های ردیابی تغییر، Change Data Capture در SQL Server چگونه کار می‌کند. در قسمت سوم در مورد ویژگی Temporal Table صحبت خواهیم کرد و این ۳ ویژگی را در کنار یکدیگر مقایسه خواهیم کرد.

راه‌حل

Change Data Capture (CDC) در SQL Server ۲۰۰۸ معرفی شد. CDC نیز مانند Change Tracking (CT)، تغییرات فعالیت DML را در یک جدول ثبت می‌کند. مانند CT، این ویژگی باید ابتدا در سطح پایگاه داده و سپس در سطح جدول فعال شود. اما برخلاف CT، CDC دارای ویژگی‌های بسیار پیچیده‌ای است. با این وجود، CDC یک ویژگی عالی است و موارد استفاده خاص خود را دارد و Temporal Table ها جایگزین مناسبی برای CDC ها نیستند.

هنگامی که یک جدول در یک پایگاه داده برای ذخیره داده‌های تغییر یافته، فعال می‌شود، تمام تغییرات آن جدول در یک جدول تغییرات، ذخیره شده و به این ترتیب تغییرات ردیابی می‌شوند. جدول تغییرات شامل یک رکورد برای هر INSERT خواهد بود که می‌تواند برای شناسایی مقادیر ستون برای رکوردهای درج شده استفاده شود. هر بار که یک DELETE انجام می‌شود، جدول تغییرات شامل یک رکورد برای هر DELETE خواهد بود که مقادیر هر ستون را قبل از DELETE نشان می‌دهد. هنگامی که یک UPDATE انجام می‌شود، در جدولی که قابلیت ذخیره داده‌های تغییر یافته را دارد، دو رکورد در جدول تغییرات ایجاد می‌شود، یکی با مقادیر ستون به روز شده و دیگری با مقادیر ستون اصلی قبل از به روز رسانی، ذخیره می‌شود. با استفاده از ذخیره داده‌های تغییر یافته، می‌توانید تغییراتی را که در طول زمان رخ داده‌اند در جدول خود ردیابی کنید. این

نوع عملکرد برای برنامه‌ها مفید است، مانند فرآیند بارگذاری انبار داده که نیاز به شناسایی تغییرات دارند، بنابراین می‌توانند به درستی به روز رسانی‌ها را برای ردیابی تغییرات تاریخی در طول زمان استفاده شوند.

ویژگی Change Data Capture در SQL Server چگونه کار می‌کند؟

یک پایگاه داده به نام DataCapture و یک جدول به نام Customer ایجاد می‌کنیم. سپس چند ردیف را در جدول Customer وارد می‌کنیم. پس از آن ویژگی CDC را در سطح پایگاه داده فعال می‌کنیم.

```
USE master
GO

CREATE DATABASE DataCapture
GO

USE DataCapture
GO

CREATE TABLE Customer (
CustomerId INT  PRIMARY KEY
,FirstName VARCHAR(۳۰)
,LastName VARCHAR(۳۰)
,Amount_purchased DECIMAL
)
GO

INSERT INTO dbo.Customer( CustomerId, FirstName,    LastName,    Amount_Purchased)
VALUES
(۱, 'Frank', 'Sinatra',۲۰۰۰۰.۰۰),( ۲,'Shawn', 'McGuire',۳۰۰۰۰.۰۰),( ۳,'Amy', 'Carlson',۴۰۰۰۰.۰۰)
GO

SELECT * FROM dbo.Customer

-- Now enable CDC at the Database Level
EXEC sys.sp_cdc_enable_db
GO
```

هنگامی که سعی کردم ویژگی CDC را در پایگاه داده DataCapture فعال کنم، یک پیام خطای بسیار دقیق دریافت کردم.

```
Msg ۲۲۸۳۰, LEVEL ۱۶, State ۱, PROCEDURE sys.sp_cdc_enable_db_internal, Line ۱۹۸ [Batch START Line ۳۱] Could
NOT UPDATE the metadata that indicates database DataCapture IS enabled FOR Change Data Capture. The failure
occurred WHEN executing the command 'SetCDCTracked(Value = ۱)'. The error returned was ۱۵۴۰۴: 'Could not
```

```
obtain information about Windows NT group/user 'NC\alalani', error code ۰x۵۴b.'. USE the action AND error TO
determine the cause OF the failure AND resubmit the request.
```

CDC مستلزم این است که owner پایگاه داده یک sysadmin باشد. به طور پیش‌فرض کاربر ایجاد کننده پایگاه داده owner خواهد بود. بنابراین تغییر owner به 'sa' خطای فوق را برطرف می‌کند و یا اگر کاربر فوق را به نقش sysadmin اضافه کنید خطا برطرف می‌شود. البته دقت داشته باشید که اضافه کردن کاربر به sysadmin همچنین مجوزهای زیادی را به کاربر می‌دهد که ممکن است نخواهید به آنها بدهید.

```
EXEC sp_changedbowner 'sa'
GO

EXEC sys.sp_cdc_enable_db
GO
```

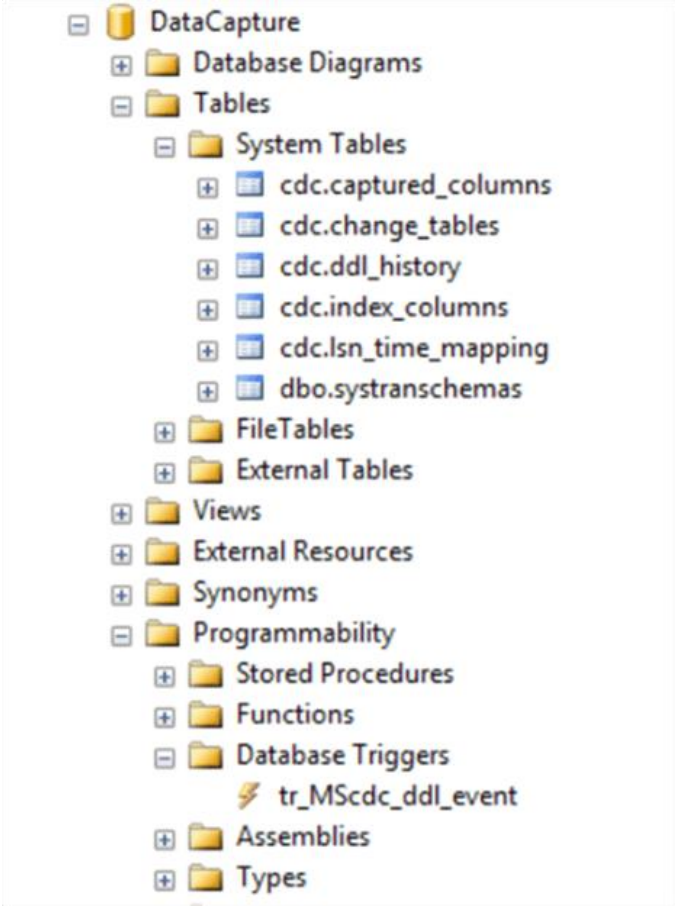
اکنون CDC را در سطح جدول فعال می‌کنیم.

```
-- Enable on the table level
EXEC sys.sp_cdc_enable_table
  @source_schema = N'dbo',
  @source_name   = N'Customer',
  @role_name     = NULL,
  @filegroup_name = N'Primary',
  @supports_net_changes = ۰
GO
```

فعال کردن CDC در سطح جدول به سادگی در سطح پایگاه داده نیست. به این دلیل است که تمام اشیا CDC به عنوان اشیا سیستم ایجاد می‌شوند. همچنین وابستگی به پایگاه داده MSDB و سرویس SQL Server Agent وجود دارد. اگر دستور بالا را با موفقیت اجرا کنیم، پیام زیر را برمی‌گرداند:

```
Job 'cdc.DataCapture_capture' started successfully.
Job 'cdc.DataCapture_cleanup' started successfully.
```

یک تریگر DDL و تعدادی از پروسیجرهای سیستم نیز ایجاد می‌شود. اشیا CDC تماما در یک پایگاه داده هستند. اگر به اشتباه جدول را drop کنید، تاریخچه شما از بین می‌رود. برخلاف Temporal Table، در صورت فعال بودن CDC، هیچ مکانیسم ایمنی برای محدود کردن حذف جدول وجود ندارد.



تغییراتی به صورت زیر در جدول Customer ایجاد می‌کنیم.

```
-- insert a row
INSERT INTO Customer (Customerid, FirstName, LastName, Amount_purchased)
VALUES (۴, 'Ameena', 'Lalani', ۵۰۰۰۰)
GO

-- delete a row
DELETE FROM dbo.Customer
WHERE CustomerId = ۲
GO

-- update a row
UPDATE Customer
SET Lastname = 'Clarkson' WHERE CustomerId = ۳
GO
```

```
-- Let us query to see what it reports
SELECT * FROM dbo.Customer

DECLARE @begin_lsn binary (۱۰), @end_lsn binary (۱۰)

-- get the first LSN for customer changes
SELECT @begin_lsn = sys.fn_cdc_get_min_lsn('dbo_customer')

-- get the last LSN for customer changes
SELECT @end_lsn = sys.fn_cdc_get_max_lsn()

-- get individual changes in the range
SELECT * FROM cdc.fn_cdc_get_all_changes_dbo_customer(@begin_lsn, @end_lsn, 'all');
```

	CustomerId	FirstName	LastName	Amount_purchased
1	1	Frank	Sinatra	20000
2	3	Amy	Clarkson	40000
3	4	Ameena	Lalani	50000

	__$start_lsn	__$seqval	__$operation	__$update_mask	CustomerId	FirstName	LastName	Amount_purchased
1	0x00000025000005B80004	0x00000025000005B80003	2	0x0F	4	Ameena	Lalani	50000
2	0x00000025000005C00005	0x00000025000005C00002	1	0x0F	2	Shawn	McGuire	30000
3	0x00000025000005C80003	0x00000025000005C80002	4	0x04	3	Amy	Clarkson	40000

توجه کنید ۲ = customerId که حذف شد، اکنون همه تغییرات ظاهر می‌شود. CDC اطلاعات تغییر DML را به صورت ناهم‌زمان می‌نویسد. ابتدا در لاگ تراکنش می‌نویسد و سپس لاگ های تراکنش را جستجو می‌کند و اطلاعات را با شروع و پایان Log Sequence Number (LSN) ذخیره می‌کند. برخلاف CT، CDC همه ستون‌ها را ذخیره می‌کند و نه فقط ستون‌های کلید اصلی. به ستون $operation در تصویر بالا نگاه کنید، عدد ۲ نشان دهنده یک Insert است، عدد ۱ برای Delete و عدد ۴ برای Update است.

بیایید ببینیم پس از تغییر ردیف‌هایی که در بالا انجام شده‌اند، اطلاعات چگونه ذخیره می‌شوند.

```
-- Update the above row one more time
UPDATE Customer
SET Lastname = 'Blacksmith' WHERE CustomerId = ۳
GO

-- Let INSERT few more rows
INSERT INTO Customer (Customerid, FirstName, LastName, Amount_purchased)
VALUES (۵, 'Sponge', 'Bob', ۵۰۰۰)
```

```
GO

INSERT INTO Customer (Customerid, FirstName, LastName, Amount_purchased)
VALUES (۶, 'Donald', 'Duck', ۶۰۰۰)
GO

-- Let us query to see what it reports now
SELECT * FROM dbo.Customer

DECLARE @begin_lsn binary (۱۰), @end_lsn binary (۱۰)

-- get the first LSN for customer changes
SELECT @begin_lsn = sys.fn_cdc_get_min_lsn('dbo_customer')

-- get the last LSN for customer changes
SELECT @end_lsn = sys.fn_cdc_get_max_lsn()

-- get individual changes in the range
SELECT * FROM cdc.fn_cdc_get_all_changes_dbo_customer(@begin_lsn, @end_lsn, 'all');
```

	CustomerId	FirstName	LastName	Amount_purchased
1	1	Frank	Sinatra	20000
2	3	Amy	Blacksmith	40000
3	4	Ameena	Lalani	50000
4	5	Sponge	Bob	5000
5	6	Donald	Duck	6000

	__$start_lsn	__$seqval	__$operation	__$update_mask	CustomerId	FirstName	LastName	Amount_purchased
1	0x00000025000005B80004	0x00000025000005B80003	2	0x0F	4	Ameena	Lalani	50000
2	0x00000025000005C00005	0x00000025000005C00002	1	0x0F	2	Shawn	McGuire	30000
3	0x00000025000005C80003	0x00000025000005C80002	4	0x04	3	Amy	Clarkson	40000
4	0x0000002500000C080003	0x0000002500000C080002	4	0x04	3	Amy	Blacksmith	40000
5	0x0000002500000C100003	0x0000002500000C100002	2	0x0F	5	Sponge	Bob	5000
6	0x0000002500000C180003	0x0000002500000C180002	2	0x0F	6	Donald	Duck	6000

Temporal Tableها در ذخیره‌سازی داده‌های تاریخی کارآمدتر هستند در مقاله بعدی به این موضوع دقیق‌تر نگاه خواهیم کرد. از آنجایی که جدول فعلی دارای داده است و آنها محدود به زمان هستند، نیازی به نگه داشتن داده‌های اضافی در شی history وجود ندارد. CDC آن داده‌های اضافی را همان طور که برای id های مشتری با شماره‌های ۵ و ۶ در مثال بالا می‌بینیم، نگه می‌دارد. CDC هیچ بُعد زمانی ندارد، تغییرات داده‌ها را بر اساس LSN (شماره توالی ورود به سیستم) پیگیری می‌کند در حالی که Temporal Table در SQL Server ۲۰۱۶ بعد زمانی را نیز در نظر می‌گیرد.

نتیجه گیری

CDC جایگاه خاص خود را دارد و Temporal Table در SQL Server ۲۰۱۶ جایگزین آنها نمی‌شوند. CDC را می‌توان برای ثبت تغییرات در Master Data در Master Data Management (MDM) با ثبت ناهم‌زمان تغییرات استفاده کرد. با مثال‌هایی در این مقاله، مزایا و معایب ویژگی CDC را مشاهده کردیم. سینتکس و اجرای کلی CDC بسیار پیچیده‌تر از Change Tracking و Temporal Table است. در مقاله بعدی ما با استفاده از یک Temporal Table به همین مثال خواهیم پرداخت و تمام ۳ ویژگی ردیابی داده در SQL Server را مقایسه خواهیم کرد.

منابع

https://www.mssqltips.com/sqlservertip/۵۲۱۲/sql-server-temporal-tables-vs-change-data-capture-vs-change-tracking-part-۲/