

بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش تکرار سازی (Replication) (قسمت سوم)

مدرس : مهندس افشین رفوآ

رمز فایل : [tahlildadeh.com](http://tahlildadeh.com)

کلیه حقوق مادی و معنوی این مقاله متعلق به آموزشگاه تحلیل داده می باشد و هر گونه استفاده غیر قانونی از آن پیگرد قانونی دارد.

### Replication ادغامی

**Replication** ادغامی جایگزین دیگری است که می توان برای سیستم های با درجه دسترس پذیری بالا استفاده کرد **Replication**. ادغامی به طور عمده برای کاربران سیار نامتصل (**disconnected**) طراحی شده. مکانیسم هایی به صورت تو کار قرار داده شده که تغییرات هر مکانی را سنکرون می کند، بعلاوه در برابر **failure** ها مقاوم بوده و پردازش را ادامه می دهد. در این بخش مکانیسم های درونی که برای **Replication** ادغامی موجود است و همچنین نحوه اعمال ان به سیستم ها را بررسی می کنیم.

### (Change Tracking)

فرایند پیگیری تغییرات در **Replication** ادغامی به این صورت است:

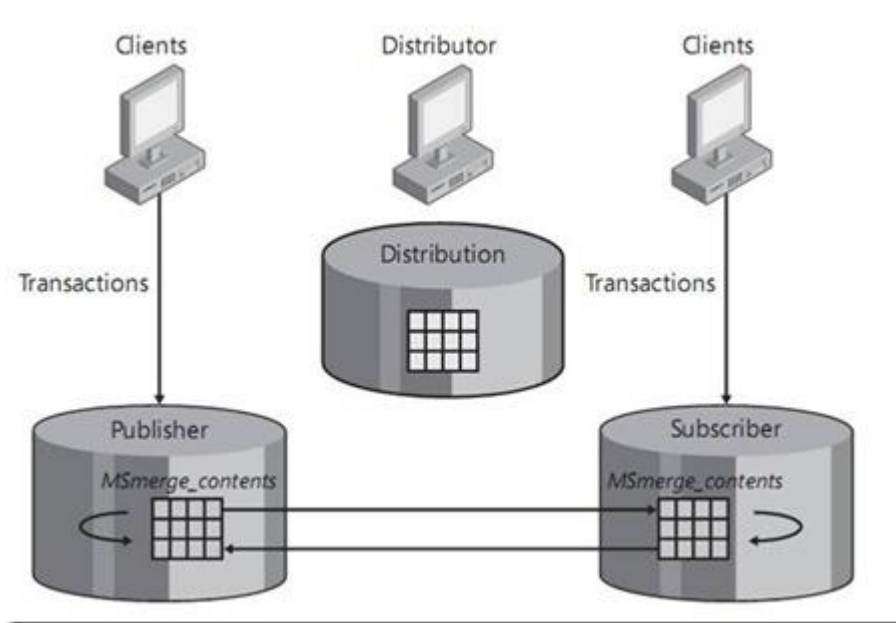
- برنامه کاربردی یک تراکنش را صادر می کند.
- **trigger** پشت جدول آغاز می شود.
- **Insert** و **update** درون **Msmerge\_contents** و **delete** درون **Msmerge\_tombstone** ثبت می شود.
- **Trigger** پایان می یابد. (**commit**)
- تراکنش پایان می یابد.

بر خلاف **Replication** تراکنشی، **Replication** ادغامی کپی های داده را به پایگاه داده توزیع انتقال نمی دهد. ثبت وقایع (**logging**) که در **MSmerge\_contents** و **MSmerge\_tombstone** رخ می دهد تنها نشانگر آن است که تغییری به یک سطر خاص اعمال شده است. داده واقعی مربوط به تغییر، کپی نمی شود؛ و فقط در جدول نگهداری می شود. این، نتیجه جالبی را در بر دارد.

اگر بین چرخه های موتور **Replication** ، **10** تراکنش مجزا برای یک سطر صادر شود و جدول با **replication** تراکنشی پیکره بندی شده باشد، تمام 10 تراکنش توسط عامل **Log Reader** به پایگاه داده توزیع انتقال می یابد، سپس هر 10 تراکنش به مشترک اعمال می شود. نتیجه اینکه فقط 9 تا از تراکنش ها اعمال می شود و سپس اثر آنها توسط تراکنش دهم جایگزین و خنثی می شود. با این حال، اگر جدول با **Replication** ادغامی پیکره بندی شده باشد، فقط آخرین وضعیت سطر (وضعیت بعد از تراکنش دهم) به مشترک منتقل می شود. برای این منظور از فرایند زیر برای سنکرون کردن تغییرات استفاده می شود:

- موتور ادغام به مشترک متصل می شود.
- موتور ادغام به منتشر کننده متصل می شود.
- موتور ادغام جدول **MSmerge\_genhistory** روی مشترک را مورد پرس و جو قرار می دهد تا لیست نسل هایی (**generations**) که در منتشر کننده موجود نیست را بیابد. (نسل یک بسته از تغییرات است که با هم اعمال می شود).
- سپس موتور ادغام، لیست جدول ها و سطرهای جدول های موجود در **MSmerge\_contents** و **MSmerge\_tombstone** را برای نسل هایی که منتشر کننده ندارد، بیرون می کشد.
- موتور ادغام دستورهای **DELETE** را از سطرهایی که از **MSmerge\_tombstone** گرفته باز سازی می کند.
- موتور ادغام محتویات سطر های جدول های **base** در پایگاه داده را استخراج و تغییرات را دسته دسته می کند.
- تغییرات دسته دسته شده به منتشر کننده اعمال می شود.
- نسل هایی که به منتشر کننده ارسال شد در **MSmerge\_contents** ثبت می شود.
- ناسازگاری ها کشف و رفع می شوند.
- موتور ادغام جدول **MSmerge\_genHistory** روی منتشر کننده را مورد پرس و جو قرار می دهد تا لیست نسل هایی که در مشترک وجود ندارد را معین کند.
- موتور ادغام سپس لیست جدول ها و سطرهای موجود در **MSmerge\_contents** و **MSmerge\_tombstone** را برای نسل هایی که مشترک ندارد، بیرون می کشد.
- موتور ادغام دستورهای **DELETE** را از سطرهایی که از **MSmerge\_tombstone** گرفته باز سازی می کند.
- موتور ادغام محتویات سطرهای جداول **base** در پایگاه داده را استخراج و تغییرات را دسته دسته می کند.
- تغییرات دسته دسته شده به مشترک اعمال می شود.
- نسل هایی که به مشترک ارسال شد در **MSmerge\_contents** ثبت می شود.

دیگرام کلی **Replication** ادغامی با فرایند سنکرون سازی در زیر آمده است.



هر بار که موتور ادغام اجرا می شود، یک درخواست از منتشر کننده و مشترک می کند: "چیزی که ندارم را به من بدهید". با وجود سادگی این درخواست، اثری قدرتمند دارد.

موتور تراکنشی تغییراتی را که باید انجام شود را ذخیره و سپس به مشترک ارسال می کند. هنگامی که یک تغییر اعمال می شود، از موتور **Replication** حذف می شود. این باعث می شود که **replication** تراکنشی فقط روی تغییراتی که در صف منتظر هستند کار کند که بدین معناست که فقط از این نقطه زمانی به جلو را می فهمد. موتور ادغام، زمانی که تغییر انجام شد و محل تغییرات را تمیز نمی دهد. موتور ادغام توانایی فهمیدن هر چیزی که در گذشته رخ داده را دارد و فقط تغییراتی که الان وجود ندارد را منتقل می کند. خلاصه، جداول متاداده (**metadata**) زیرین موتور ادغام، مانند یک رکورد پرواز است که تغییرات پایگاه داده را از زمانی که آن رکورد (موتور ادغام) روشن شده، توضیح می دهد.

همین پروسه است که موتور ادغام را اینقدر انعطاف پذیر کرده است. تمام تغییرات در **MSmerge\_contents** ، **MSmerge\_gethistory** و **MSmerge\_tombstone** ثبت می شود. مهم نیست تغییر توسط موتور **replication** انجام شده یا توسط یک برنامه کاربردی؛ جدول های متاداده ادغامی، همه چیز را ثبت می کنند.

یکی از مشکلات موتورهای تراکنشی این است که تغییرات در گستره سه کمپوننت مجزا پخش شده است **log** : تراکنش روی منتشر کننده، پایگاه داده توزیع و مشترک . (**subscriber**) بنابراین، به شدت مشکل است که در حالی که **Replication** فعال است از وضعیت سیستم بتوان پشتیبان گرفت و یا سیستم را بازیابی (**restore**) کرد.

**replication** ادغامی این مشکل را نیز حل کرده است. جدول های متاداده ادغام در همان پایگاه داده ای قرار دارند که **article** های منتشر شونده شما قرار دارد. بنابراین، هنگامی که از پایگاه داده پشتیبان می گیرید، همزمان از متاداده ادغام نیز پشتیبان می گیرید؛ ان متاداده کاملا با محتویات جداول سنکرون می باشد. شما بعدا می توانید هر **backup** از پایگاه داده را از زمانی که **publication** ادغامی **replicate** کردن تغییرات را شروع کرده، بازیابی کنید و همه کارها را به عهده موتور ادغام بگذارید. مهم نیست تغییرات از کجا شروع شده، موتور ادغام خیلی ساده نسل هایی که روی منتشر شونده یا مشترک وجود ندارد را یافته و همه چیزهایی که وجود ندارد را ارسال می کند. لازم نیست درباره بازیابی سیستم به وضعیتی خاص نگران باشید موتور ادغام ترتیب همه کارها را می دهد.

طی بحث پیگیری تغییرات و سنکرون سازی به کرات از نسل نام بردیم. جدول های متاداده ی ادغام، حاوی تاریخچه کاملی از تمام سطوری است که از زمان آغاز **replication** ادغامی تغییر کرده اند. پرس و جوی متاداده برای تغییرات به صورت سطر به سطر به شدت ناکارا است. برای کاهش سربار، موتور ادغام تغییرات را به دسته هایی به نام نسل (**generation**) بسته بندی می کند که به حالت همه-یا-هیچ بین منتشر کننده و مشترک اعمال می شود. موتور **replication** می تواند بر اساس نسل بدنبال تغییرات روی منتشر کننده یا مشترک بگردد.

اعتبارسنجی یک **publication** ادغامی خیلی شبیه به اعتبارسنجی **replication** تراکنشی است. اعتبارسنجی را می توان در دو حالت انجام داد: فقط شمارش سطر و شمارش سطر بعلاوه **checksum** باینری. عملیات و سربار مانند **replication** تراکنشی است.

برای اعتبارسنجی یک **publication** کامل میتوانید **sp\_validate merge publication** و برای اعتبارسنجی فقط یک مشترک می توانید **sp\_validate merge subscription** را اجرا کنید، و یا می توانید آپشن **-Validate** را به عامل ادغام اضافه کنید.

## Replication ادغامی

در این تمرینها، با استفاده از پلیگاه داده **AWMerge** یک **Replication** ادغامی را پیکره بندی می کنیم.

### تمرین اول: ساخت یک publication ادغامی

یک پایگاه داده به نام **AWMergeSubscriber** در همان نمونه ای که پایگاه داده ای که **AWMerge** روی ان است، بسازید.

۱. در **Object Browser**، روی گره **Replication** کلیک راست مرده و **New Publication** را انتخاب کنید. روی **Next** کلیک کنید.
۲. پایگاه داده **AWMerge** را انتخاب و روی **Next** کلیک کنید.
۳. **Merge Publication** را انتخاب و روی **Next** کلیک کنید.
۴. مطمئن شوید **SQL Server 2008** انتخاب شده و روی **Next** کلیک کنید.

۵. تمام جدول ها و UDF ها را انتخاب کنید، همانطوری که هنگام ساخت publication تراکنشی انجام دادید. روی Next کلیک کنید.
۶. دوباره روی Next کلیک کنید. فیلترینگ انجام نخواهیم داد. دوباره روی Next کلیک کنید.
۷. مطمئن شوید که Create A Snapshot Immediately انتخاب شده و تیک گزینه Schedule the Snapshot Agent To Run At The following Times روی Next کلیک کنید.
۸. تنظیمات امنیتی را همانند publication تراکنشی مشخص کنید. روی Next کلیک کنید.
۹. مطمئن شوید Create the Publication انتخاب شده باشد. روی Next کلیک کنید.
۱۰. برای publication تان یک نام انتخاب کنید و Finish را کلیک کنید. بعد از اینکه publication ساخته شد روی Close کلیک کنید.
۱۱. پایگاه داده AWMerge را بررسی کنید، تا تمام تغییراتی که به triggerها، روال های ذخیره شده، ساختار جدول ها، ایندکس ها و view ها اعمال شده تا از پیگیری تغییر در replication ادغامی پشتیبانی کنند، را ببینید.

## تمرین ۲: ساخت یک Subscription ادغامی

۱. Local Publication را بسط دهید، روی publication که بالا ساختید کلیک راست کرده و New Subscriptions را انتخاب کنید.
۲. publication ادغامی تان را انتخاب و روی Next کلیک کنید.
۳. مطمئن شوید گزینه Run All Agents As The Distributer انتخاب شده و روی Next کلیک کنید.
۴. چک باکس کنار نمونه تان را انتخاب و پایگاه داده AWMergeSubscriber را از لیست پایی افتادنی Subscription Database انتخاب کنید.
۵. دکمه ... کنار مشترکتان را در پنجره Subscription Property بزنید.
۶. گزینه Run Uner The SQL Server Agent Service Account را انتخاب و هر دو گزینه By Impersonating The Process Account. OK و بعد روی Next کلیک کنید.
۷. در صفحه Synchronization Schedule ، گزینه Run Continuously را از لیست پایین افتادنی Agent Schedule انتخاب و روی Next کلیک کنید.
۸. مطمئن شوید مقدار Subscription Properties برابر Initialize Immediately است بعد روی Next کلیک کنید.
۹. مقدار Subscription Type Of Server را پیش فرض نگه دارید و روی Next کلیک کنید.
۱۰. مطمئن شوید گزینه Create Subscription(s) انتخاب شده باشد. روی Next کلیک کنید. و بعد Finish.
۱۱. بعد از ساخته شدن Subscription روی Close کلیک کنید.
۱۲. با استفاده از Replication Monitor ، عامل ها، تاریخچه، و پیغام های خطا را بررسی کنید.

۱۳. پایگاه داده **AwMergeSubscriber** را بررسی کنید تا ببینید بعد از اعمال **Snapshot** چه اشیائی ساخته شده.

۱۴. منتشر کننده و مشترک را تغییر دهید تا ببینید چگونه داده ها در موتور حرکت می کند.

۱۵. ناسازگاری داده ای تولید کنید و نحوه کشف و جل انرا مشاهده کنید.

• موتور ادغامی موتوری پرقدرت و انعطاف پذیر برای توزیع تغییرات است چون برای کاربرای سیار و نامتصل طراحی شده است.

• از آنجا که موتور ادغام طوری نوشته شده تا از پس قطعی های زیاد و سنکرون داده های غیرمنتظره بر بیاید، می توان **specification** انرا که برای کاربران سیار و نامتصل طراحی شده، در محیط های سرور به سرور به کار برد.

• پیگیری تغییرات در یک مجموعه از جداول متاداده درون همان پایگاه داده ای که در حال انتشار و یا اشتراک است، رخ می دهد.

• از آنجا که هم منتشر کننده و هم مشترک یک کپی کامل از تمام تغییرات انجام شده را نگه می دارند، موتور ادغام فقط یک درخواست ساده در هر چرخه می کند: "هر آنچه را که ندارم برابم بفرستید."