

병합 복제와 피어 투 피어 복제

(주)엑셈 컨설팅본부/SQL Server팀 이 제춘

복제의 개요

데이터베이스 복제란 데이터베이스의 개체들을 다른 데이터베이스에 복사 한 후 동기화하여 일관성을 유지하는 기술이다. 아래 그림과 같이 서버 과부하를 분산시키기 위해 또는 데이터의 보존 및 개발, 분석 등을 위해 복제를 사용하기도 한다. 이렇게 여러 가지 목적으로 데이터베이스 복제를 하는데 이런 목적들에 따라 복제 도 몇 가지 유형으로 나뉘게 된다.

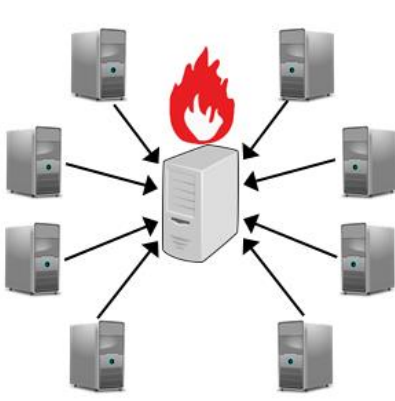


그림 1. 단일 데이터베이스 부하

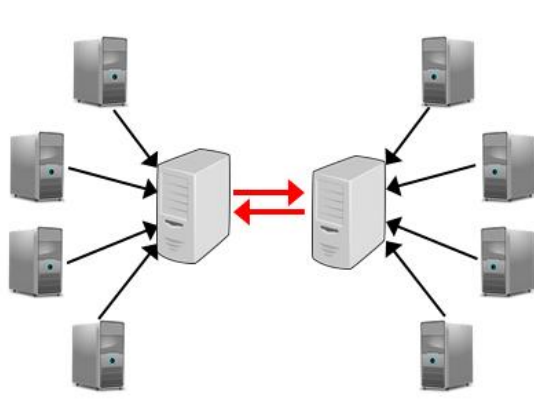


그림 2. 데이터베이스 복제를 통한 분산

크게 스냅샷 복제, 트랜잭션 복제, 병합 복제로 나뉘는데 지금 다루고자 하는 내용 중 피어 투 피어 복제는 트랜잭션 복제에 속한다. SQL Server 2012 부터는 복제 구성 시 피어 투 피어 복제도 하나의 게시 유형으로 스냅샷 복제, 트랜잭션 복제, 병합 복제, 피어 투 피어 복제 이렇게 4개의 게시유형이 있다.

스냅샷 복제와 트랜잭션 복제는 많이 사용 되고 있고 참고 할만한 텍스트도 많이 있는 반면 병합 복제와 피어 투 피어 복제는 비교적 알려지지 않았고 참고자료 또한 빈약 하다. 이 두 가지의 복제는 서로 닮은 점도 많은데 이 부분은 뒤에서 언급하도록 하겠다. 데이터베이스 미러링과 클러스터링 그리고 SQL Server 2012 에서 발표된 Always On 까지 성능 좋은 고 가용성 솔루션들 틈에서 병합복제와 피어 투 피어 복제의 포지션을 찾아 보고자 한다.

복제의 기초

병합 복제와 피어 투 피어 복제에 대한 설명에 앞서 복제에 대한 기본 바탕을 알아야겠다. 복제 구성 시 포함되는 서버의 역할과 구성요소, 그리고 먼저 언급했던 복제의 유형에 대해 알아보자.

복제 구성에 대해 설명 할 때 출판사 혹은 잡지사를 빗대 예를 들곤 한다. 복제 구성요소들의 명칭이 출판업계에서 사용되는 말들과 같기 때문이다. 복제에 이용되는 서버는 게시자(Publisher), 배포자(Distributor), 구독자(Subscriber) 이렇게 세가지 역할의 서버가 있다. 이 서버들이 복제하는 대상인 데이터베이스 개체(테이블, 뷰, 저장 프로시저 등)를 아티클(Article)이라 하고 복제 동작에 대해 게시(Publication)와 구독(Subscription)이라고 한다. 이렇게 6 가지가 복제의 구성요소라고 할 수 있다.

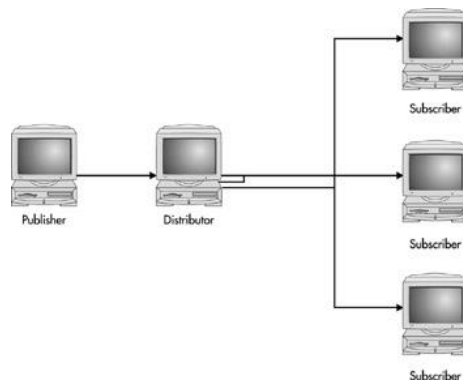


그림 3. 복제 구성의 예

위 그림의 구조는 가장 먼저 게시자 서버가 배포자 서버에게 아티클을 게시하고 배포자 서버는 3 대의 구독자 서버에 아티클을 배포하는 구조이다. 기본 구조는 이렇지만 실제로는 다른 경우가 많다. 경우에 따라 다르겠지만 배포자 서버를 따로 두지 않고 게시자 혹은 구독자와 함께 구성

할 수도 있다. 일반적인 경우 배포자가 겹치게 되는 구성(한 서버가 두 개의 역할)에 대해 추천하고 싶지 않는데 이 같은 경우 배포자를 함께 갖고 있는 서버에게 무리가 될 수 있기 때문이다. 물론 부하가 현저히 낮고 복제 양이 매우 적은 경우 등 때에 따라 배포자를 다른 역할의 서버와 함께 구성하여 서버의 개수를 줄여 금전적인 부담을 더는 방법을 사용 할 수도 있다.

복제는 기본적으로 3 가지 유형이 있는데 스냅샷 복제, 트랜잭션 복제, 병합 복제가 그것이다. 아래 세가지 복제를 간단히 비교해 보았다.

스냅샷 복제
- 게시에서 지정한 모든 개체 및 데이터의 복사본을 만든다. - 데이터의 업데이트를 모니터링 하지 않는다. - 단독으로 사용 가능하지만 트랜잭션 및 병합 게시 이전 초기동기화를 위해서도 사용된다.

스냅샷 복제는 많은 양의 데이터가 변경되지만 자주 변경되지 않을 때 적합하다. 데이터 업데이트를 모니터링 하지 않기 때문에 연속적인 오버헤드가 적은 반면 스냅샷 복제가 수행될 때 변경 내역만이 아닌 게시에서 설정 한 모든 데이터를 복제하기 때문에 많은 리소스가 필요하게 된다. 이 부분을 고려하여 스냅샷 복제를 사용하는 것이 좋다.

트랜잭션 복제
- 일반적으로 스냅샷 복제를 통해 게시의 모든 데이터를 복사 한 이 후 수행된다.⁴ - 스냅샷 이 후 후속 데이터 변경 내용 및 스키마 수정 내용이 거의 실시간으로 수행된다. - 게시 내에서는 트랜잭션 일관성이 보장된다.

트랜잭션 복제는 거의 실시간으로 변경 내용이 구독자로 전파되어야 할 때 사용 한다. 빈번한 삽입, 업데이트, 삭제 등의 DML 이 있으며 게시와 구독이 거의 실시간으로 동기화 되어 일관성을

⁴ 게시 할 아티클이 구독자 서버에 동일하게 구성되어 있어야 하는데 백업, 복원 등 의 방법들 보다 스냅샷 복제를 이용하는 것이 편리하기 때문에 일반적으로 스냅샷 복제를 이용한다.

유지해야 할 경우 사용된다. 기본적으로 트랜잭션 게시의 구독자는 읽기전용으로 취급하지만 옵션을 통해 구독자의 업데이트를 허용 할 수 있다. 그러나 이는 추천하고 싶지 않다.⁵

병합 복제

- 트랜잭션 복제와 마찬가지로 스냅샷 복제를 시작으로 수행된다.
- 서버-클라이언트 간 환경에 주로 사용 된다.
- 구독자는 네트워크에 연결될 때 게시와 동기화하여 마지막 동기화 이후 변경된 모든 행을 교환한다.

병합 복제는 여러 사이트에서 작업 한 내용이 취합되어 하나의 일관된 데이터로 동기화 된다. 서버-클라이언트 간 환경에 주로 사용되는 이유가 여기에 있다. 병합 과정에서 충돌이 발생 할 수 있지만 병합복제는 충돌을 처리하는 다양한 방법을 제공한다.

병합복제와 피어 투 피어 복제

본격적으로 병합복제와 피어 투 피어 복제를 비교해 보자. 앞서 복제유형에 대해 설명하면서 간략하게 병합복제에 대해 알아보았다. 여기서는 피어 투 피어 복제에 대해 먼저 알아보자.

피어 투 피어 복제

- 트랜잭션 복제 유형에 속하며 트랜잭션 복제에 대한 게시 유형의 하나이다.
- 여러 노드의 데이터가 거의 실시간으로 유지 관리된다.
- 데이터 중복을 제공하며 중복으로 인해 가용성을 높여준다.

⁵ 트랜잭션 복제는 충돌처리가 빈약하다. 양쪽이 자주 변경을 해야 한다면 충돌처리가 좋은 병합복제를 적용하는 것이 좋다. ‘트랜잭션 복제는 구독자에서의 업데이트를 허용하는 옵션을 제공하지만, 일반적으로 권고되지는 않으며, 언젠가는 없어질 기능이다.’ – 이장래 저, 『SQL Server 2012 운영과 개발』에서 발췌

피어 투 피어 복제는 모든 노드가 게시자이자 구독자가 되는 형태 이다. 피어 투 피어를 이용하여 읽기 작업 효율을 높일 수 있지만 쓰기성능은 단일 노드와 같다. 이는 모든 insert, update, delete 와 같은 DML 작업이 한 서버에서 이루어지면 곧바로 모든 노드에 똑같이 전파되기 때문이다.

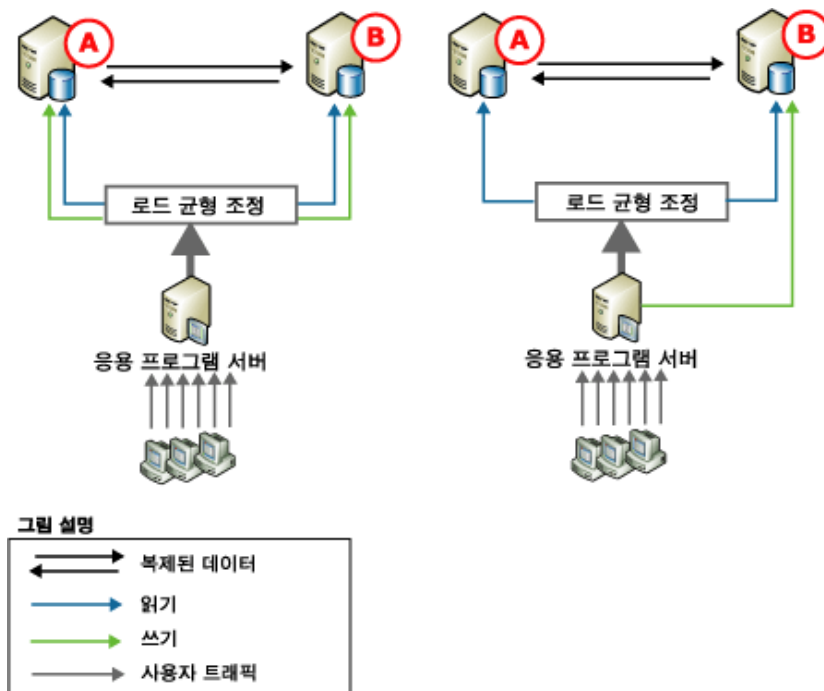


그림 4. 쓰기 작업이 각각 다른 두 개의 토폴로지 피어 투 피어 복제

위 그림은 두 개의 토폴로지 구성일 때 쓰기작업이 분산되는 경우와 한 곳에서만 이루어지는 경우를 나타낸 그림이다. 좌측의 경우 양쪽 모두에 쓰기작업을 하고 있는데 이럴 경우 충돌 가능성이 존재한다. 피어 투 피어 복제에는 두 가지 방법이 모두 지원되지만 표준 트랜잭션 복제에서는 우측과 같이 한곳에서만 업데이트를 하도록 하는 방식이 주로 쓰인다.

그럼 병합 복제와 피어 투 피어 복제를 직접 비교해 보도록 하겠다.

	병합 복제	피어 투 피어 복제
복제 유형	병합 복제 유형	트랜잭션 복제 유형
충돌 가능성	있음	있음
충돌 시 대안	충돌 감지 및 해결	제한적(충돌 검색)
동기화	에이전트 일정에 따라 ⁶	거의 실시간
게시	중앙 게시	구성된 모든 노드
중복 저장	중복 저장	중복 저장

두 복제의 가장 큰 차이점으로 볼 수 있는 것이 충돌 시 대안과 동기화 시간이다. 가상환경에 두 복제를 구성하여 확인해 본 결과 피어 투 피어는 1 초도 안 걸리는 시간으로 거의 실시간에 가깝게 동기화 하고 병합 복제는 일정에 따라 동기화 된다. 일정은 복제 구성 시 에이전트 일정에서 설정 할 수 있다. 가장 단시간으로 설정 할 경우 10 초마다 동기화 한다. 아래에서는 충돌 관련 내용을 정리해 보고 실제 어떻게 동작하는지 테스트 해보도록 하겠다.

병합 복제와 피어 투 피어 복제에서의 충돌

병합 복제는 충돌 감지와 충돌을 해결하는 다양한 방법이 있다. MS에서는 대부분의 응용프로그램에 기본 방법이 가장 적합하다고 권장하고 있다. 기본 방법은 다음과 같다.

10. 게시, 구독 간 충돌 발생시 게시자 변경 내용이 유지되고 구독자 변경내용은 삭제된다.
11. 끌어오기 구독을 하는 두 구독자 사이 충돌 발생 시 게시자와 동기화하는 첫 번째 구독자의 변경내용만 유지되고 두 번째 구독자의 변경 내용은 삭제된다.
12. 밀어 넣기 구독을 사용하는 두 구독자 사이 충돌이 발생하면 우선순위가 높은 구독 변경 내용이 유지되고 두 번째 구독자의 변경 내용은 삭제된다. 우선 순위가 같으면 2 번과 같이 먼저 동기화 하는 구독자의 변경 내용이 유지된다.

⁶ 에이전트 일정을 선택 할 수 있다. 요청 시에만 실행과 계속 실행 중 선택 가능하고 일정 정의를 통해 주기를 정할 수 있다. 병합 복제에서 최단 동기화 시간은 10 초에 한번씩 동작 하는 것이 가장 최단 시간이다.

병합 복제에는 충돌 감지 및 해결을 위한 고급 기능들이 더 있지만 본 서에서는 기본 방법만을 다루도록 하겠다.

피어 투 피어 복제는 병합 복제에 비해 충돌 시 대응 방법이 빈약하다. 게시 속성 대화 상자의 구독 옵션 페이지나 토폴로지 구성 페이지를 사용하여 충돌을 검색 할 수 있도록 설정할 수 있는데⁷ 충돌에 대한 검색만 가능하고 확인 후 관리자가 수동으로 일련의 조치를 해야 한다. 충돌이 검색된 노드를 정상 노드의 백업에서 다시 초기화 하는 방법이 권장되는데 배포 에이전트에서 변경 내용 적용을 계속하도록 하여 노드를 다시 동기화 해야 한다.

병합 복제와 피어 투 피어 복제에서 충돌을 발생시키고 이 충돌을 검색하고 처리하는 방법에 대해 직접 테스트 해 보도록 하겠다.

병합 복제의 충돌 시나리오

병합 복제 충돌 테스트 구성환경은 다음과 같다.

서버 : 배포&게시 서버1, 구독 서버2

구독 : 끌어오기 구독

일정 : 매 1분 1회(계속 실행 Default)

아티클 : 테이블 1개[

	no	name	dep	score	rowguid
1	1	홍길동	총무팀	12345	8C8120BB-CCCB-E211-81EF-00155D7BBF08
2	2	장발장	영업팀	10000	755593E5-CFCB-E211-81EF-00155D7BBF08
3	3	피터팬	개발팀	20000	A0982AEE-CFCB-E211-81EF-00155D7BBF08
4	4	뽀로로	개발팀	30000	73EE4DF8-CFCB-E211-81EF-00155D7BBF08

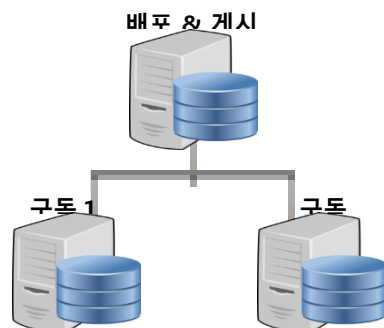


그림 5. 병합 복제 테스트 구성

⁷ SQL Server 2008 이상 버전에서는 SSMS 에서 기본적으로 충돌 검색이 사용되도록 설정되어 있다.

가장 흔하게 일어 날 수 있는 충돌인 update-update 충돌을 발생 시켰다. 구독 1 에서 장발장의 score 를 50000 으로 update 한 후 동기화 되기 전 게시에서도 20000 으로 update 를 해 주었다. 동기화 일정이 되기 전 까지는 장발장의 score 가 게시는 20000, 구독 1 은 50000, 구독 2 는 기존의 10000 으로 데이터 불일치가 발생한다. 동기화가 진행되며 충돌이 발생하게 되지만 지금과 같은 게시-구독 충돌의 경우 기본적으로 게시 쪽 데이터가 적용되었다. 복제 충돌 뷰어를 통해 발생 한 충돌을 확인 할 수 있고, 여기서 충돌에 대한 조치를 취할 수 있다.

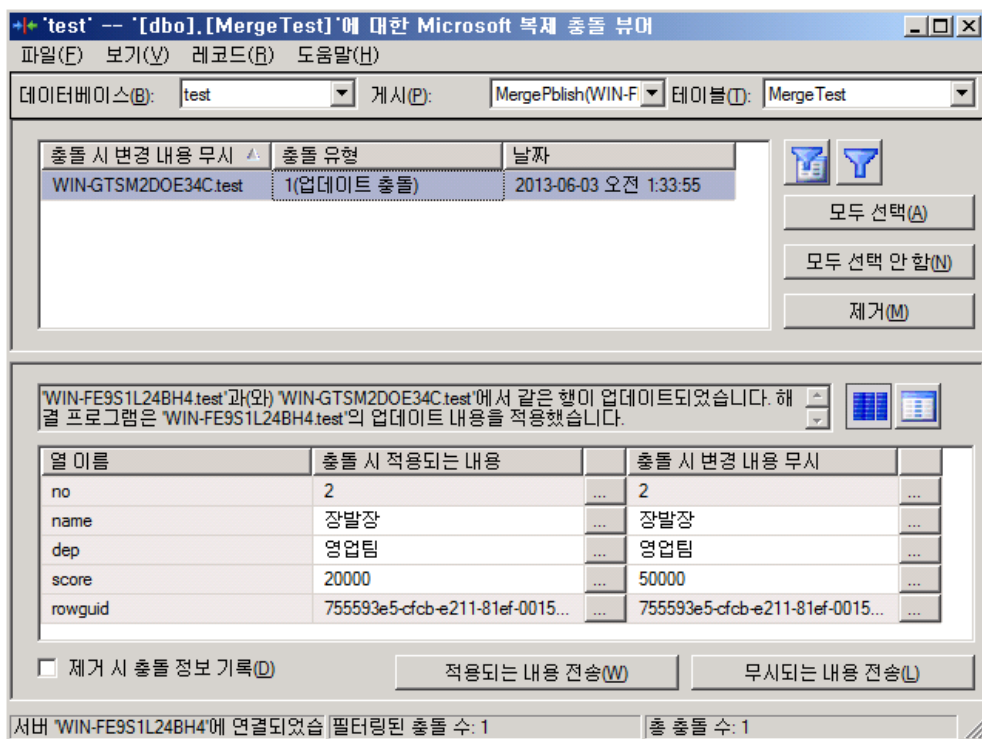


그림 6. 복제 충돌 뷰어

구독 1 과 구독 2 에서도 update-update 충돌을 발생 시켜 보았는데 여기에서는 먼저 실행 한 update 가 자동으로 적용되는 것을 확인 할 수 있었다. 끌어오기 구독에서는 먼저 변경 된 데이터가 실행되고 나중에 실행 된 변경은 무시되었다. 밀어넣기 구독의 경우에는 구독에 대해 우선 순위를 적용 할 수 있는 옵션이 적용되어 우선순위에 따라 충돌에 대해 변경이 적용 된다. 충돌 유형이 다른 update-delete 는 실행 순서에 관계 없이 update 를 수행하였다. 재미있는 것은 insert 인데 no 열이 identity 로 primary key 가 설정된 상태에서 각 구독에서 동시에 insert 를

할 경우 같은 번호를 부여 받아 충돌이 발생할 것이라 생각했다. 게시 쪽에서 한자리 단위의 identity 번호가 노드 1에서는 2000 번대 노드 2에서는 4000 번대 identity 를 부여 받아 insert-insert 충돌이 발생하지 않도록 구성 되었다. 이렇듯 병합 복제에서는 충돌 발생 시 자동으로 유연한 대처를 하며 또, 복제 충돌 뷰어를 통해 충돌에 대한 내용을 확인하고 직접 수동으로도 조치 할 수 있도록 구성 되었다.

피어 투 피어 복제의 충돌 시나리오

다음은 피어 투 피어 복제의 충돌 시나리오이다. 병합 복제와 거의 흡사한 환경을 구성하였다.

서버 : 모든 서버(3) 각각 배포&게시&구독

구독 : 밀어넣기 구독

피어 : 3개(피어 투 피어 토폴로지 구성 사용)

아티클 : 테이블 1개

	no	name	dep	score
1	1	홍길동	총무팀	12345
2	2	장발장	영업팀	10000
3	3	피터팬	개발팀	20000
4	4	뽀로로	개발팀	30000

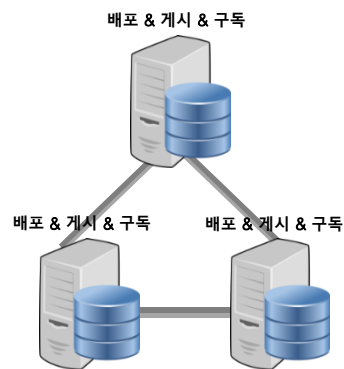


그림 7. 피어 투 피어 복제 테스트 구성

병합 복제와 다르게 rowguid 열이 없다. 구성 시 스냅샷 복제과정이 따로 없어 수동으로 아티클을 동일하게 맞춰주어야 한다. 배포는 한군데에서만 사용 하도록 구성 할 수 있지만 MS 의 권장 사항은 단일 배포를 지양하는 것이다. 원격 배포자가 단일 오류 지점일 수 있기 때문이다.

피어 투 피어 복제에서는 delete-update 충돌을 발생시켜 보았다. 4 번 Row 를 1 번 Peer 에서 삭제시키면서 거의 동시에 3 번 Peer 에서 업데이트 시켰다. 충돌이 발생 했으며 충돌 이후 양 쪽 데이터의 일관성이 깨졌다. 1 번 Peer에서는 삭제 된 데이터가 3 번 Peer에서는 삭제 되지 않고 수정되어 있었으며 그 후에 발생하는 DML 은 동기화 되지 않고 따로 수행 되었다. 2 번 Peer 를 확인 해 본 바 1 번 Peer 와 동기화 되어 있었고 3 번 Peer 의 구독에서 동기화 상태를 확인 한 바 아래와 같이 충돌을 확인 할 수 있었다.

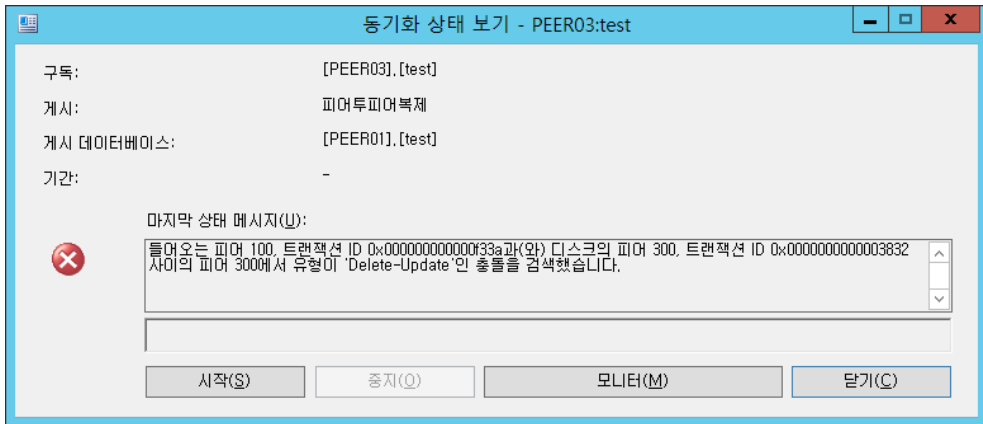


그림 8. Delete-Update 충돌

안타깝게도 충돌에 대한 검색만을 제공 할 뿐 기본적으로 자동조치가 없어 불편하다. 복제 충돌 뷰어에서도 확인 할 수 없어 조치가 불가능 하다. 충돌이 일어난 Peer 에서 복제모니터를 확인 하면 동기화 실패 메시지를 확인 할 수 있다. 충돌이 일어나지 않은 Peer 의 복제모니터에는 충돌이 발생한 3 번 Peer 가 제외 된 1, 2 번 Peer 만 복제모니터에 나타난다.

해결 방법으로 충돌이 일어나지 않은 Peer 에서 수동으로 충돌이 일어난 3 번 Peer 에 백업 및 복원하여 다시 1 번, 2 번 Peer 와 동일하게 재 구성 해 주고 다시 동기화 해 주는 방법이 권장된다. 권장 방법 이외 방법으로 게시의 속성을 충돌 감지 시 배포 에이전트에서 변경 내용을 계속 처리하도록 변경하고 배포에이전트를 시작하면 충돌 뷰어에서 충돌에 대한 확인이 가능해진다.

아래는 게시 이름이 '피어투피어복제'인 게시의 속성을 변경하여 충돌 시 배포 에이전트 변경 내용을 계속 처리하도록 하는 예시이다. 충돌을 모두 해결 하면 속성을 FALSE 로 다시 변경해주어야 한다.⁸

```
EXEC sp_changepublication '피어투피어복제', 'p2p_continue_onconflict', TRUE
```

⁸ 기본값인 FALSE 를 사용하는 것이 좋다. TRUE 일 경우 배포 에이전트는 송신자 ID 의 가장 높은 노드에서 충돌 행을 적용하여 토폴로지의 데이터를 일치시킨다. 이 방법으로 데이터가 일치하게 되지 않는 경우도 있다. 충돌이 검색된 후 토폴로지의 일관성을 확인해야 한다. – MSDN , 라이브러리 참조

충돌 뷰어를 통해 충돌에 대한 동기화를 진행하고 유효성 검사를 하여 모든 노드가 동기화 될 경우 토폴로지 구성을 확인하는 방법으로 충돌을 해결 할 수 있다.

피어 투 피어 복제는 충돌 자체가 발생하지 않게끔 설정 하는 것이 중요하다고 MSDN⁹에서 확인할 수 있다. 한 곳 에서만 쓰기 작업을 하도록 하거나 각각 다른 영역에 접근 가능하도록 설정 해야 한다. 이를테면 30 의 공간이 배정되어 있다면 1~10 은 Peer 1 번만 접근하고 11~20 은 2 번, 21~30 은 3 번이 각 각 쓰기 작업을 수행 하도록 응용 프로그램 단에서 지정해 주거나 각각 다른 시간대에 서버를 이용하도록 하는 방법을 MS 에서 권장하고 예로 들고 있다.

병합 복제와 피어 투 피어 복제 정리

주제에 대해 선정 하고 자세히 알아보기 전 까지는 병합 복제와 피어 투 피어 복제의 다른 점에 대한 인지가 부족했다. 여러 노드의 서버가 서로 데이터를 주고받아 동기화 한다는 것에만 주안 점을 두고 봤던 것 같다. 중앙서버와 지점간의 동기화가 필요하고 굳이 데이터를 실시간으로 유지 할 필요가 없는 경우 병합 복제를 사용 하는 것이 옳은 판단일 것이다.

문제는 피어 투 피어 복제인데 권장사항 대로 한쪽 Peer 에만 쓰기 작업을 하는 경우는 SQL Server 2012 의 새로운 고 가용성 기능인 Always On 의 등장으로 그다지 효용을 느낄 수 없다. 피어 투 피어 복제는 Always On 과 마찬가지로 Enterprise 이상 에서만 사용이 가능하기 때문이다. 피어 투 피어 복제를 사용 하는 경우는 실시간으로 여러 서버에서 읽기 쓰기를 사용 할 경우로 구성 설계 시 충돌 자체가 일어나지 않도록 설계하는 것이 가장 중요 할 것이다.

새로운 고 가용성 기능들이 꾸준히 나타남에도 복제의 필요성은 아직 많다. 병합 복제와 피어 투 피어 복제 또한 각각 기능에 맞는 상황에서 단점을 커버하고 사용 한다면 충분히 유용하게 사용 될 수 있는 좋은 기능임에 틀림없다. 본 서를 통해 복제의 기본, 병합 복제와 피어 투 피어 복제의 장단점과 차이점을 알고 필요한 상황에 유용하게 사용 할 수 있었으면 좋겠다.

⁹ Microsoft 가 제공하는 제품 및 온라인 설명서 등이 있다. (MicroSoft Developer Network)
한국어 주소는 다음과 같다. <http://msdn.microsoft.com/ko-kr/default.aspx>