

Microsoft®  
**SQL Server™ 2005**

**SQL Server 2005 의 데이터베이스 미러링**

본 문서는 예비 문서이며 여기에서 설명된 소프트웨어의 최종 상용 버전이 출시되기 전에 상당 부분 변경될 수 있습니다. 이 문서에 포함된 정보는 문서 발행 시에 논의된 문제들에 대한 Microsoft Corporation의 당시 관점을 나타냅니다. Microsoft는 변화하는 시장 상황에 부응해야 하므로 이를 Microsoft 측의 공약으로 해석해서는 안되며 발행일 이후 소개된 어떠한 정보에 대해서도 Microsoft는 그 정확성을 보증하지 않습니다.

이 문서는 오직 정보를 제공하기 위한 것입니다. Microsoft는 이 설명서에서 어떠한 명시적이거나 묵시적인 보증도 하지 않습니다. 해당 저작권법을 준수하는 것은 사용자의 책임입니다. 저작권에서의 권리와는 별도로, 이 설명서의 어떠한 부분도 Microsoft의 명시적인 서면 승인 없이는 어떠한 형식이나 수단(전기적, 기계적, 복사기에 의한 복사, 디스크 복사 또는 다른 방법) 또는 목적으로도 복제되거나, 검색 시스템에 저장 또는 도입되거나, 전송될 수 없습니다.

Microsoft가 이 설명서 본안에 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산권 등을 보유할 수도 있습니다. 서면 사용권 계약에 따라 Microsoft로부터 귀하에게 명시적으로 제공된 권리 이외에, 이 설명서의 제공은 귀하에게 이러한 특허권, 상표권, 저작권 또는 기타 지적 재산권 등에 대한 어떠한 사용권도 허여하지 않습니다.

특별한 언급이 없는 한, 용례에 사용된 회사, 기관, 제품, 도메인 이름, 전자 메일 주소, 로고, 사람, 장소, 이벤트 등은 실제 데이터가 아닙니다. 어떠한 실제 회사, 기관, 제품, 도메인 이름, 전자 메일 주소, 로고, 사람, 장소 또는 이벤트와도 연관시킬 의도가 없으며 그렇게 유추해서도 안됩니다.

© 2005 Microsoft Corporation. 전원 보유.

Microsoft와 ActiveX는 미국, 대한민국 및/또는 기타 국가에서 Microsoft의 등록 상표 또는 상표입니다. 여기에 인용된 실제 회사와 제품 이름은 해당 소유자의 상표일 수 있습니다.

# Contents

|                                                    |    |                                   |    |
|----------------------------------------------------|----|-----------------------------------|----|
| 들어가는 글 .....                                       | 2  | 시나리오 HASL3.1: 감시 서버 손실 .....      | 28 |
| 데이터베이스 미러링 개요 .....                                | 2  | 시나리오 HASL3.2: 감시 서버 손실에 이어        |    |
| 작동 모드 .....                                        | 3  | 주 서버 손실 .....                     | 28 |
| 트랜잭션 안전성 .....                                     | 3  | 시나리오 HASL3.3: 감시 서버 손실에 이어 미러     |    |
| 쿼럼 및 감시 서버 .....                                   | 4  | 서버 손실 .....                       | 29 |
| 고가용성 작동 모드 .....                                   | 4  | 요약: 서버 손실 시의 고가용성 시나리오 .....      | 29 |
| 높은 수준의 작동 모드 .....                                 | 5  | 통신 손실 시의 고가용성 시나리오 .....          | 31 |
| 고성능 작동 모드 .....                                    | 5  | 시나리오 HACL1.2: 주/미러 서버 회선이 끊기고     |    |
| 데이터베이스 스냅샷 및 미러 데이터베이스 .....                       | 6  | 이어서 주/감시 서버 회선이 끊김 .....          | 33 |
| 클라이언트측 측정 지표 .....                                 | 6  | 요약: 서버 손실 시의 고가용성 시나리오: 세 개의      |    |
| Database Mirroring 및 SQL Server 2005 Edition ..... | 7  | 사이트 .....                         | 34 |
| Database Mirroring Dynamics .....                  | 8  | 시나리오 HACL4: 미러 사이트에 감시 서버가 있는     |    |
| 설정 및 보안 .....                                      | 9  | 두 사이트 .....                       | 35 |
| Database Mirroring 카탈로그 뷰 .....                    | 11 | 시나리오 HACL5: 주 사이트에 감시 서버가 있는      |    |
| Database Mirroring 상태 및 전환 .....                   | 14 | 두 사이트 .....                       | 36 |
| 주 서버 데이터베이스 상태 .....                               | 15 | 요약: 통신 링크가 끊긴 상태의 고가용성 시나리오 ..... | 37 |
| 미러 서버 데이터베이스 상태 .....                              | 16 | 높은 수준의 보호 시나리오 .....              | 37 |
| 감시 서버 상태 .....                                     | 17 | 고가용성 시나리오 .....                   | 38 |
| 트랜잭션 로그 레코드 전송 .....                               | 17 | 데이터베이스 미러링 구현 .....               | 40 |
| 데이터베이스 미러링 역할 변경 .....                             | 20 | 데이터베이스 미러링 모니터링 .....             | 40 |
| 자동 장애 조치 .....                                     | 20 | 데이터베이스 미러링 Perfmon 카운터 .....      | 40 |
| 수동 장애 조치 .....                                     | 21 | 재실행 및 추적 시간 예측 .....              | 41 |
| 강제 서비스(Forced Service) .....                       | 21 | Profiler 이벤트 .....                | 41 |
| 데이터베이스 미러링 가용성 시나리오 .....                          | 22 | 데이터베이스 미러링 문제 해결 .....            | 41 |
| 서버 손실 시의 고가용성 시나리오 .....                           | 23 | 설치 문제 해결 .....                    | 41 |
| 시나리오 HASL1.1: 주 서버 손실 .....                        | 23 | 실행 시간 오류 문제 해결 .....              | 42 |
| 시나리오 HASL1.2: 주 서버 손실에 이어 새로운                      |    | 안정성 대 성능 .....                    | 42 |
| 주 서버 손실 .....                                      | 24 | 데이터베이스 미러링 테스트 .....              | 43 |
| 시나리오 HASL1.3: 주 서버 손실에 이어 감시                       |    | 미러 서버의 장애 조치 준비 .....             | 43 |
| 서버 손실 .....                                        | 25 | 다중 데이터베이스 문제 .....                | 44 |
| 시나리오 HASL2.1 미러 서버 손실 .....                        | 26 | 데이터베이스 미러링 및 고가용성 기술 .....        | 45 |
| 시나리오 HASL2.2: 미러 서버 손실에 이어                         |    | 데이터베이스 미러링 및 클러스터링 .....          | 46 |
| 주 서버 손실 .....                                      | 26 | 데이터베이스 미러링 및 트랜잭션 복제 .....        | 46 |
| 시나리오 HASL2.3: 미러 서버에 이어 감시                         |    | 데이터베이스 미러링 및 로그 전달 .....          | 47 |
| 서버 손실 .....                                        | 27 | 결론 .....                          | 48 |

## 들어가는 글

데이터베이스 미러링은 데이터베이스 가용성을 증가시키기 위한 새로운 SQL Server 2005 기술입니다. 데이터베이스 미러링은 트랜잭션 로그 레코드를 한 서버에서 다른 서버로 직접 전송하고 대기 서버로 신속하게 장애 조치할 수 있습니다. 클라이언트 응용 프로그램이 연결 정보를 자동으로 재전송하고 장애 조치의 경우 대기 서버와 데이터베이스에 자동으로 연결하도록 코딩할 수 있습니다.

데이터 손실을 최소화하는 빠른 장애 조치는 전통적으로 하드웨어 비용이 높고 소프트웨어 복잡성이 커집니다. 그러나 데이터베이스 미러링은 커밋된 데이터 손실이 없이 빠르게 장애 조치할 수 있으며, 전용 하드웨어가 필요하지 않고 설치와 관리가 쉽습니다.

## 데이터베이스 미러링 개요

데이터베이스 미러링에서 원래 SQL Server 2005 인스턴스는 다른 대기 SQL Server 인스턴스에 있는 데이터베이스 복사본으로 데이터베이스 트랜잭션 로그 레코드를 지속적으로 보냅니다. 원본 데이터베이스와 서버에는 주 서버의 역할이 있으며 수신 데이터베이스와 서버에는 미러 서버의 역할이 있습니다. 주 서버와 미러 서버는 SQL Server 2005의 개별 인스턴스가 되어야 합니다.

모든 SQL Server 데이터베이스에서 실제 데이터 페이지를 변경하기 전에 트랜잭션 로그에 데이터 변경이 기록됩니다. 트랜잭션 로그 레코드는 먼저 메모리의 데이터베이스 로그 버퍼에 저장되었다가 가능한 빨리 디스크에 플러시(또는 '하드엔디드')됩니다. 데이터베이스 미러링에서 주 서버가 주 데이터베이스의 로그 버퍼를 디스크에 쓰는 것과 동시에 로그 레코드 블록을 미러 인스턴스로 보냅니다.

미러 서버가 로그 레코드 블록을 받으면 먼저 로그 레코드를 미러 데이터베이스의 로그 버퍼에 저장한 다음 가능한 빨리 디스크에 저장하여 보호합니다. 이러한 트랜잭션 로그 레코드는 나중에 미러에서 재생됩니다. 미러 데이터베이스는 주 트랜잭션 로그 레코드를 재생하기 때문에 주 데이터베이스에 데이터베이스 변경을 복제합니다.

주 서버와 미러 서버는 각각 데이터베이스 미러링 세션의 파트너로 간주됩니다. 데이터베이스 미러링 세션은 한 파트너에서 다른 파트너로 데이터베이스를 미러링할 때 파트너 서버 사이의 관계로 구성됩니다. 해당 파트너 서버는 한 데이터베이스에는 주 서버 역할을 갖고 다른 데이터베이스에서는 미러 서버 역할을 가질 수 있습니다.

두 파트너 서버(주 서버 및 미러 서버) 이외에 데이터베이스 미러링 세션은 감시라고 하는 옵션인 세 번째 서버를 가질 수 있습니다. 감시 서버의 역할은 자동 장애 조치를 수행하는 것입니다. 고가용성을 위해 데이터베이스 미러링을 사용하면 갑자기 주 서버에 장애가 발생하고, 미러 서버가 감시 서버에서 확인을 받은 경우 자동으로 주 서버의 역할을 맡고 몇 초 내에 데이터베이스를 사용할 수 있도록 만들 수 있습니다.

데이터베이스 미러링에 대해 유의할 몇 가지 중요한 사항은 다음과 같습니다.

- 주 데이터베이스는 FULL 복구 모델에 있어야 합니다. 대량 로그 작업의 로그 레코드는 미러 데이터베이스로 보낼 수 없습니다.
- 미러 데이터베이스는 NORECOVERY를 사용하여 주 데이터베이스의 복원에서, 이어서 순차적으로 주 트랜잭션 로그 백업의 복원으로부터 초기화해야 합니다.
- 미러 데이터베이스 이름은 주 데이터베이스 이름과 같아야 합니다.
- 미러 데이터베이스가 복구 중인 상태에 있기 때문에 직접 액세스할 수 없습니다. 미러에 데이터베이스 스냅샷을 만들어 특정 시간에 미러 데이터베이스를 간접적으로 읽을 수 있습니다. 이 백서 뒷부분의 '데이터베이스 미러링 및 데이터베이스 스냅샷'을 참조하십시오.

주: 데이터베이스 미러링과 관련된 용어에 대한 자세한 내용은 SQL Server 2005 온라인 설명서의 "Overview of Database Mirroring"을 참조하십시오.

## 작동 모드

데이터베이스 미러링 세션에 대해 세 가지 가능한 작동 모드가 있습니다. 정확한 모드는 트랜잭션 안전 및 감시 서버가 미러링 세션의 일부인지 여부를 기초로 합니다.

표 1. 데이터베이스 미러링 작동 모드

| 작동 모드     | 트랜잭션 안전 | 전송 메커니즘 | 필요한 쿼럼 | Witness 서버 | 장애 조치 유형 |
|-----------|---------|---------|--------|------------|----------|
| 고가용성      | FULL    | 동기식     | Y      | Y          | 자동 또는 수동 |
| 높은 수준의 보호 | FULL    | 동기식     | Y      | N          | 수동만      |
| 높은 성능     | OFF     | 비동기     | N      | 없음         | 강제만      |

안전이 FULL인 경우 비동기 데이터 전송이 발생하고 데이터베이스 서비스를 위해 쿼럼이 필요합니다. 쿼럼 투표에는 각각의 두 파트너 서버가 재생해야 하는 역할, 주 또는 미러를 결정하기 위해 최소 두 서버가 필요합니다.

세 개의 작동 모드를 자세히 알아 보려면 먼저 트랜잭션 안전과 쿼럼 역할을 자세히 살펴 보십시오.

## 트랜잭션 안전성

트랜잭션 안전성(또는 그냥 '안전성')이 FULL로 설정된 경우 주 서버와 미러 서버는 동기 전송 모드로 작동합니다. 기본 서버가 주 데이터베이스 로그 레코드를 디스크에 저장할 때 해당 레코드를 미러 서버에도 보냅니다. 주 서버는 미러 서버의 응답을 기다립니다. 같은 로그 레코드를 미러 서버의 로그 디스크에 저장하면 미러가 응답합니다. 안전성이 OFF로 설정되면 기본 서버는 미러로부터 승인을 기다리지 않으므로 미러는 완전히 동기화되지 않을 수 있습니다. 즉, 미러가 주 서버와 동일한 상태를 유지하지 못할 수 있습니다.

동기 전송은 미러 데이터베이스의 트랜잭션 로그에 있는 모든 트랜잭션이 주 데이터베이스의 트랜잭션 로그와 동기화될 수 있도록 하므로 트랜잭션은 안전하게 전송된 것으로 간주됩니다. 다음을 사용하여 안전성을 FULL로 설정합니다.

```
ALTER DATABASE [<dbname>] SET SAFETY FULL;
```

안전성이 OFF로 설정되면 주 서버와 미러 사이의 통신이 동기화됩니다. 주 서버는 미러가 트랜잭션 레코드의 블록을 저장한 미러에서 승인되는 것을 기다리지 않습니다. 미러는 가능한 빨리 트랜잭션을 기록함으로써 주 서버와 동일한 상태를 유지하려고 하지만 주 서버에 갑자기 장애가 발생하여 미러가 서비스하도록 하는 경우 일부 트랜잭션이 손실될 수 있습니다. SQL Server 온라인 설명서의 'Forced Service' 항목을 참조하십시오.

## 쿼럼 및 감시 서버

안전성이 FULL이면 데이터베이스 미러링 세션은 쿼럼을 사용하여 데이터베이스 서비스를 유지합니다. 쿼럼은 동기 데이터베이스 미러링 세션에서 필요로 하는 모든 연결된 서버 사이에 최소 관계를 나타냅니다. 쿼럼에는 최소 두 서버가 필요하기 때문에 안전성이 FULL이면 주 서버는 데이터베이스의 서비스를 유지하기 위해 최소 하나의 다른 서버와 쿼럼을 구성해야 합니다.

감시 서버는 쿼럼 구성에서 주 서버나 미러를 지원합니다. 감시 서버가 있는 경우 주 데이터베이스나 미러 데이터베이스 중 하나가 손실되면 두 서버가 쿼럼을 구성하게 됩니다. 주 서버가 미러 서버에 연결할 수 없지만 감시 서버와 쿼럼을 구성할 수 있는 경우 데이터베이스의 서비스는 유지할 수 있습니다. 마찬가지로, 미러와 감시 서버가 주 서버에 연결할 수 없고 미러 서버가 감시 서버와 쿼럼을 구성할 수 없는 경우 미러가 새로운 주 서버의 역할을 맡을 수 있습니다.

감시 서버에 장애가 발생하는 경우 주 서버와 미러 서버가 계속 쿼럼을 구성하기 때문에 감시 서버는 데이터베이스 미러링 세션의 단일 장애 지점으로 간주되지 않습니다. 자세한 내용은 SQL Server 온라인 설명서의 "Quorum in Database Mirroring Sessions"를 참조하십시오.

## 고가용성 작동 모드

고가용성 작동 모드는 주 데이터베이스에 장애가 발생하는 경우 미러 데이터베이스로 자동 장애 조치하여 최대한의 데이터베이스 가용성을 지원합니다. 이렇게 하려면 안전성을 FULL로 설정하고 감시 서버를 데이터베이스 미러링 세션의 일부로 정의해야 합니다.

서버 간에 빠르고 매우 안정적인 통신 경로를 갖는 고가용성 모드를 사용하는 것이 가장 좋으며 단일 데이터베이스에 대한 자동 장애 조치가 필요합니다. 안전성이 FULL이면 주 서버는 잠시 동안 미러 서버로부터 응답을 기다려야 하므로 미러 서버의 기능이 주 서버의 성능에 영향을 줄 수 있습니다. 단일 데이터베이스 오류로 인해 자동 장애 조치되기 때문에 여러 데이터베이스 응용 프로그램이 있는 경우 다른 작동 모드를 고려할 수 있습니다. 이 문서 뒷부분에 있는 구현 절의 "다중 데이터베이스 문제"를 참조하십시오.



고가용성 모드에서 데이터베이스 미러링은 자체 미러링입니다. 주 데이터베이스를 갑자기 사용할 수 없게 되거나 주 서버가 다운되는 경우 감시 서버와 미러 서버는 쿼럼을 구성하며 미러 SQL Server는 자동 장애 조치를 수행합니다. 이 지점에서 미러 서버 인스턴스는 새로운 주 서버로 역할이 변경되고 데이터베이스를 복구합니다. 미러가 주 서버의 트랜잭션 로그를 재생하고 주 서버와 동기화했기 때문에 미러 서버를 신속하게 사용할 수 있습니다.

또한 SQL Server 2005는 복구 프로세스 초기 단계에서 사용자가 데이터베이스를 사용하도록 만들 수 있습니다. SQL Server 데이터베이스 복구는 분석 단계, 재실행 단계 및 마지막으로 실행 취소 단계 등 세 단계로 구성됩니다. SQL Server 2005에서 새로 복구된 데이터베이스는 재실행 단계가 완료되는 대로 사용할 수 있습니다. 따라서 데이터베이스 미러링 장애 조치가 발생하는 경우 재실행 단계를 완료하는 즉시 복구된 새로운 주 데이터베이스를 사용할 수 있습니다. 미러 데이터베이스가 트랜잭션 로그 레코드를 완전 재생했기 때문에 미러 서버가 수행해야 하는 일은 재실행 프로세스를 완료하는 것입니다. 이는 일반적으로 몇 초 만에 수행될 수 있습니다.

## 높은 수준의 작동 모드

높은 수준의 작동 모드에서는 트랜잭션 안전성이 FULL이지만 미러링 세션의 일부로 감시 서버가 없습니다. 주 데이터베이스는 여전히 쿼럼을 구성해야 하지만 감시 서버가 없어 미러 서버만으로 구성해야 합니다. 이 모드에서는 타이 브레이커(tie breaker) 역할을 충족시키는 감시 서버가 없기 때문에 수동 장애 조치만 가능합니다. 주 서버에 장애가 발생하는 경우 미러 서버에는 쿼럼을 구성할 감시 서버가 없기 때문에 자동 장애 조치는 가능하지 않습니다.

안전성이 FULL이면 주 서버와 미러의 쿼럼이 갑자기 상실될 경우 데이터베이스 서비스를 중단해야 합니다. 감시 서버를 일시적으로 제거해야 할 때는 고가용성의 경우 전환 상태로 데이터베이스 미러링 구성을 제외하므로 높은 수준의 보호 모드는 사용하지 않는 것이 좋습니다.

## 고성능 작동 모드

고성능 작동 모드에서는 트랜잭션 안전성이 OFF이고 로그 레코드 전송이 비동기화됩니다. 주 서버는 모든 트랜잭션 로그 레코드가 미러에 기록된 미러에서 승인되기를 기다리지 않습니다. 미러는 주 서버와 같은 상태를 유지하지만 어느 한 시점에 주 서버의 모든 최신 트랜잭션이 미러의 트랜잭션 로그에 저장될 것임을 보장하지 않습니다.

고성능 모드에서는 감시 서버는 아무 역할도 하지 않으며 쿼럼이 필요하지 않습니다. 따라서 자동 및 수동 장애 조치는 고성능 모드에 사용되지 않습니다. 허용되는 유일한 장애 조치 유형은 수동 작업이라고도 하는 forced service 장애 조치입니다.

```
ALTER DATABASE <dbname> SET PARTNER FORCE_SERVICE_ALLOW_DATA_LOSS
```

강제 서비스 장애 조치를 수행하면 미러 데이터베이스가 즉시 복구됩니다. 주 서버의 일부 트랜잭션 로그 블록을 미러에서 아직 받지 못한 경우 복구되면 미러에서 잠재적인 데이터 손실이 발생할 수 있습니다. 고성능 모드는 먼 거리로 데이터를 전송하거나(즉, 원격 사이트로 재해 복구하는 경우), 약간의 잠재적 데이터 손실을 허용하는 매우 활성화된 데이터베이스를 미러링하는 경우 사용하면 매우 좋습니다.

## 데이터베이스 스냅샷 및 미러 데이터베이스

미러 데이터베이스가 복구 중인 상태에 있기 때문에 액세스하거나 읽을 수 없습니다. SQL Server 2005 Enterprise Edition 및 Developer Edition에서는 데이터베이스 스냅샷을 만들어 어느 한 시점에 미러 데이터베이스를 읽을 수 있습니다. 데이터베이스 스냅샷은 데이터베이스의 읽기 전용 보기를 제공하여 스냅샷을 만드는 시점에 일관성 있는 데이터를 표시합니다.

마치 다른 데이터베이스에 있는 것처럼 데이터베이스 스냅샷에 액세스합니다. 데이터베이스 스냅샷을 쿼리하면 데이터베이스 스냅샷 파일에서 스냅샷을 만든 후에 변경된 데이터베이스 데이터의 원본 버전을 읽고 원본 데이터베이스에서 변경되지 않은 데이터를 읽습니다. 결과적으로 스냅샷을 만든 시점에 최신 데이터베이스 데이터를 볼 수 있습니다. 자세한 내용은 SQL Server 온라인 설명서의 “Using Database Snapshots with Database Mirroring”을 참조하십시오.

데이터베이스 스냅샷은 미러 서버에 약간의 오버헤드가 요구되기 때문에 데이터베이스 미러링 성능에 어떤 영향을 미칠 수 있는지에 주의해야 합니다. 한 데이터베이스로만 미러링할 수 있으므로 많은 읽기 전용 보고 서버로 확장해야 하는 경우 트랜잭션 복제가 더 나은 선택이 될 것입니다. 자세한 내용은 뒷부분의 “데이터베이스 미러링 및 복제”를 참조하십시오.

## 클라이언트측 측정 지표

SQL Server 2005에서 ADO.NET 또는 SQL Native Client로 미러링하는 데이터베이스에 연결하는 경우 응용 프로그램은 데이터베이스 미러링 장애 조치가 발생할 때 자동으로 연결을 리디렉션하는 드라이버 기능을 이용할 수 있습니다. 연결 문자열에서 초기 주 서버와 데이터베이스를, 선택적으로 장애 조치 파트너 서버를 지정해야 합니다.

연결 문자열을 작성하는 방법은 많지만 여기에서 제시하는 한 예는 서버 A를 주 서버, 서버 B를 미러 및 AdventureWorks를 데이터베이스 이름으로 지정합니다.

```
“Data Source=A;Failover Partner=B;Initial Catalog=AdventureWorks;Integrated Security=True;”
```

연결 문자열에서 장애 조치 파트너는 초기 주 서버 연결이 실패하는 경우 대체 서버 이름으로 사용됩니다. 초기 주 서버 연결이 성공하면 장애 조치 파트너 이름은 사용되지 않지만 드라이버는 클라이언트측 캐시에 있는 주 서버에서 검색하는 장애 조치 파트너 이름을 저장합니다.

클라이언트가 주 서버에 성공적으로 연결되었고 데이터베이스 미러링 장애 조치(자동, 수동 또는 강제)가 발생한다고 가정합니다. 다음에 응용 프로그램이 연결을 사용하려고 시도하면 ADO.NET 또는 SQL Native Client 드라이버는 이전 주 서버 연결이 실패한 것을 발견하고 장애 조치 파트너 이름에 지정된 새로운 주 서버에 자동으로 연결을 재시도합니다. 성공하고 새로운 주 서버에 의해 데이터베이스 미러링 세션에 지정된 새 미러 서버가 있는 경우 드라이버는 새로운 파트너 장애 조치 서버 이름을 검색하고 클라이언트 캐시에 저장합니다. 클라이언트가 대체 서버에 연결할 수 없는 경우 드라이버는 로그인 시간 초과 기간에 도달할 때까지 각 서버에 연결을 시도합니다.

ADO.NET 및 SQL Native Client 드라이버에 기본 제공되는 데이터베이스 미러링 지원을 사용하는 큰 장점은 데이터베이스 미러링 장애 조치를 처리하기 위해 응용 프로그램을 다시 코딩하거나 응용 프로그램에 특별한 코드를 삽입할 필요가 없다는 것입니다.

ADO.NET 또는 SQL Native Client 자동 리디렉션을 사용하지 않을 경우 응용 프로그램을 장애 조치할 수 있는 다른 기술을 사용할 수 있습니다. 예를 들어, 네트워크 로드 균형 조정을 사용하여 한 서버에서 다른 서버로 연결을 수동으로 리디렉션할 수 있지만 클라이언트는 가상 서버 이름에 연결합니다. 또한 자체 리디렉션 코드를 작성하고 로직을 다시 시도할 수 있습니다.

그러나 데이터베이스 미러링을 사용하여 클라이언트 리디렉션을 통합하는 이러한 모든 기술에는 중요한 제한이 있습니다. 데이터베이스 미러링은 서버 수준이 아니라 데이터베이스 수준에서만 발생합니다. 응용 프로그램이 서버의 여러 데이터베이스 쿼리에 의존하거나 여러 부분의 개체 이름을 사용하여 여러 데이터베이스에 쿼리하는 경우 주의하십시오. 한 서버에 여러 데이터베이스가 있고 대기 서버에 미러링되면 여러 데이터베이스 중 하나가 대기 서버로 장애 조치될 수 있지만 다른 데이터베이스는 원래 서버에 남아 있습니다. 이 경우 쿼리하는 데이터베이스 당 하나의 연결이 필요할 수 있으므로 한 데이터베이스만 주 서버이고 나머지는 미러인 대기 서버에서 데이터베이스간 쿼리를 시도하지 마십시오.

## Database Mirroring 및 SQL Server 2005 Edition

다음 표는 SQL Server 2005의 다양한 버전에서 어떤 데이터베이스 미러링 기능이 지원되는지 보여줍니다.

표 2. 데이터베이스 미러링 및 SQL Server 2005 Edition

| Database Mirroring<br>주요 특징 | Enterprise<br>에디션 | Developer<br>에디션 | Standard<br>에디션 | Workgroup<br>에디션 | SQL Express |
|-----------------------------|-------------------|------------------|-----------------|------------------|-------------|
| 파트너                         | ✓                 | ✓                | ✓               |                  |             |
| Witness                     | ✓                 | ✓                | ✓               | ✓                | ✓           |
| 안전성 = FULL                  | ✓                 | ✓                | ✓               |                  |             |
| 안전성 = OFF                   | ✓                 | ✓                |                 |                  |             |
| 장애 조치 후 UNDO<br>동안 사용 가능    | ✓                 | ✓                | ✓               |                  |             |
| 병렬 재실행                      | ✓                 | ✓                |                 |                  |             |
| 데이터베이스<br>스냅샷               | ✓                 | ✓                |                 |                  |             |

몇 가지 데이터베이스 미러링 기능은 SQL Server 2005 Enterprise or Developer Edition이 필요합니다.

- 안전성이 OFF인 고성능 모드(비동기 데이터 전송);
- 데이터베이스 스냅샷;
- 미러 데이터베이스에서 트랜잭션 로그를 재생하는 데 여러 스레드 사용(병렬 REDO).

SQL Express 및 Workgroup Edition을 감시 서버로 사용할 수 있지만 데이터베이스 미러링에서는 파트너 서버로 사용할 수 없습니다.

## Database Mirroring Dynamics

SQL Server 2005 데이터베이스 미러링을 자세히 이해하려면 데이터베이스 미러링 세션이 시간이 지남에 따라 어떻게 변하는지 확인하는 것이 도움이 됩니다. 이 절에서는 데이터베이스 미러링 세션의 여러 가지 데이터베이스 상태, 동기 및 비동기 로그 레코드 전송의 메커니즘, 장애 조치 순서를 다룰 것입니다.

### 설정 및 보안

주 서버, 미러 서버 및 선택적으로 감시 서버를 식별한 후에 Database Mirroring 설정은 세 단계로 구성됩니다. 데이터베이스를 백업하고 복구 없이 원하는 미러 서버로 복원해야 합니다.

**주:** 데이터베이스를 백업하여 미러 서버에 복원하기 전에 주 데이터베이스는 FULL 복구 모델에 있어야 합니다. 트랜잭션 로그에서 대량 로그 레코드를 전송해야 하는 경우 데이터베이스 미러링이 작동하지 않습니다. 미러 서버에는 주 데이터베이스만큼 파일이 확장될 충분한 여유 공간이 있어야 합니다. 미러에 데이터베이스 스냅샷을 만들려면 추가 여유 공간을 제공해야 합니다.

백업, 복사 및 복원에 상대적으로 오랜 시간이 걸리는 경우 로그 크기를 제어하기 위해 원본 데이터베이스에서 트랜잭션 로그 백업을 수행해야 합니다. 그러나 트랜잭션 로그 레코드가 로그 백업에 의해 로그에서 잘린 경우 데이터베이스 미러링을 초기화하지 못할 수 있습니다. 따라서 데이터베이스 미러링을 초기화하기 전에 복구 없이 원하는 미러 데이터베이스로 이러한 트랜잭션 로그 백업을 순서대로 복원해야 합니다.

데이터베이스 미러링 세션에 관련된 서버를 서로를 신뢰해야 합니다. 도메인 같은 로컬 통신의 경우 신뢰는 각 SQL Server 인스턴스 로그인에 다른 미러링 서버에 연결할 권리가 있어야 하며 해당 중점을 수행하는 것을 의미합니다. 각 서버에서 먼저 CREATE LOGIN 명령을 사용한 다음 GRANT CONNECT ON ENDPOINT 명령을 사용하여 이를 설정합니다. SQL Server 온라인 설명서의 "Example of Setting Up Database Mirroring Using Windows Authentication"을 참조하십시오.

신뢰하지 않는 도메인을 통해 통신하는 경우 인증서를 사용해야 합니다. CREATE CERTIFICATE 문을 사용하여 자체 서명된 인증서를 만드는 경우 대부분의 데이터베이스 미러링 인증서 요구 사항이 만족됩니다. 또한 CREATE CERTIFICATE 문에서 인증서가 ACTIVE FOR BEGIN\_DIALOG로 표시되었는지도 확인해야 합니다.

다음 단계는 데이터베이스 미러링 종단을 설정하는 것입니다. 종단점을 설정하려면 SQL Server 인스턴스에 대해 시스템 관리자 권한이 필요합니다. 데이터베이스 미러링 종단점으로 명시적으로 만든 각 서버에서 종단점을 설정해야 합니다. 종단점을 설정하는 가장 간단한 방법은 Configure Database Mirroring Security Wizard를 사용하는 것입니다. Database Properties 대화 상자의 Mirroring 페이지에서 Configure Security 단추를 클릭하여 실행할 수 있습니다. Configure Security 대화 상자는 CREATE ENDPOINT 명령을 구성하고 실행하기 전에 컴퓨터 이름과 포트 번호 및 선택적으로 로그인을 묻습니다. Transact-SQL을 사용하여 CREATE ENDPOINT 명령을 실행할 수도 있습니다. SQL Server 온라인 설명서의 "How to: Create a Mirroring Endpoint (Transact-SQL)"를 참조하십시오.



도메인에 데이터베이스 미러링을 설정하고 있고 모든 SQL Server 인스턴스가 같은 서비스 로그인과 암호를 사용하는 경우 각 서버에 로그인을 만들 필요는 없습니다. 마찬가지로 작업 그룹에서 모든 SQL Server 인스턴스가 같은 서비스 로그인과 암호를 사용하는 경우 서버에 로그인을 만들 필요가 없습니다. 종단점을 설정할 때 Configure Database Mirroring Security Wizard에서 로그인을 비워두면 됩니다.

각 데이터베이스 종단점은 서버에 고유한 포트를 지정해야 합니다. 개별 시스템에서 SQL Server 인스턴스 작업을 수행하면 이러한 포트 번호는 모두 같을 수 있으며 Configure Database Mirroring Security Wizard는 포트 5022를 자동으로 제안합니다. SQL Server 인스턴스가 같은 시스템에 있는 경우 각 인스턴스는 개별 포트를 가져야 하며 포트 번호는 고유해야 합니다.

고가용성 미러링 세션에서 세 개의 서버를 사용한다고 가정합니다. 서버 A는 주 서버, 서버 B는 미러, 서버 W는 감시 서버가 됩니다. 서버 A의 경우 다음 명령은 포트 5022에 종단점을 만듭니다.

```
CREATE ENDPOINT [Mirroring]
AS TCP (LISTENER_PORT = 5022)
FOR DATABASE_MIRRORING (ROLE = PARTNER, ENCRYPTION = ENABLED);
```

역할이 PARTNER로 지정되었으므로, 이 서버는 해당 데이터베이스 미러링 데이터베이스에 대해 주 또는 미러 서버의 역할을 맡을 수 있습니다. 서버 B에 같은 명령이 실행됩니다. 서버 B는 개별 실제 시스템에 있는 SQL Server 인스턴스이므로 포트 번호는 동일합니다. 서버 W의 경우 다음 명령을 실행할 수 있습니다.

```
CREATE ENDPOINT [Mirroring]
AS TCP (LISTENER_PORT = 5022)
FOR DATABASE_MIRRORING (ROLE = WITNESS, ENCRYPTION = ENABLED);
```

서버 W의 경우 역할은 WITNESS로 지정되었습니다.

기본적으로 종단점은 시작되지 않습니다. 다음에 각 서버에서 다음 쿼리를 사용하여 각 종단점을 시작할 수 있습니다.

```
ALTER ENDPOINT [Mirroring] STATE = STARTED;
```

선택적으로 CREATE ENDPOINT 명령에 STATE 옵션을 삽입할 수 있습니다. 이 프로세스는 각 서버에서 반복됩니다. CREATE ENDPOINT를 사용하여 종단점을 만들 때 프로토콜 관련 인수를 사용하여 IP 주소로 액세스를 제한할 수 있습니다. RESTRICT\_IP를 ALL 옵션과 함께 사용하고 EXCEPT\_IP를 원하는 특정 IP 주소 목록과 함께 사용하여 IP 주소의 특정 집합에 대한 액세스를 제한할 수 있습니다. SQL Server 온라인 설명서의 “CREATE ENDPOINT”를 참조하십시오.

sys.database\_mirroring\_endpoints 카탈로그 뷰를 쿼리하여 서버에서 데이터베이스 미러링 종단점을 검사할 수 있습니다.

```
SELECT *
FROM sys.database_mirroring_endpoints;
```

4. 데이터베이스 미러링을 시작하려면 파트너와 감시를 지정합니다. 해당 데이터베이스 미러링 세션을 시작하고 관리하려면 데이터베이스 소유자 권한이 있어야 합니다. 원하는 주 서버인 서버 A에서 SQL Server가 특정 데이터베이스에 주 서버 역할을 부여하고 어느 것을 파트너(미러) 서버로 할 것인지 지정하도록 합니다.

-- 주 서버에서 파트너 지정

```
ALTER DATABASE [AdventureWorks] SET PARTNER =  
N'TCP://B.corp.mycompany.com:5022' ;
```

파트너 이름은 파트너의 정규화된 컴퓨터 이름이어야 합니다. 정규화된 이름을 찾는 것은 어려울 수 있지만 중단점을 설정하면 Configure Database Mirroring Security Wizard가 이를 자동으로 찾아줍니다.

각 서버의 정규화된 컴퓨터 이름은 명령 프롬프트에서 다음 명령을 실행하여 찾을 수도 있습니다.

```
IPCONFIG /ALL
```

"Host Name" 및 "Primary DNS Suffix"를 연결합니다. 다음과 같은 내용이 표시되는 경우

```
Host Name . . . . . : A  
Primary Dns Suffix . . . . . : corp.mycompany.com
```

그러면 컴퓨터 이름은 A.corp.mycompany.com이고 접두사 'TCP://' 및 접미사는 ':<port number>'이며 파트너 이름이 있습니다.

미러 서버에서는 같은 명령을 반복하지만 주 서버 이름은 다음과 같습니다.

-- 미러 서버에서 파트너 지정

```
ALTER DATABASE [AdventureWorks] SET PARTNER =  
N'TCP://A.corp.mycompany.com:5022' ;
```

주 서버에서 감시 서버를 지정합니다.

-- 주 서버에서 감시 서버 지정

```
ALTER DATABASE [AdventureWorks] SET WITNESS =  
N'TCP://W.corp.mycompany.com:5026' ;
```

초기 CREATE ENDPOINT 다음에 감시 서버에서 추가 명령을 실행할 필요는 없습니다.

마지막으로 주 서버에서 세션의 안전성을 지정합니다.

-- 주 서버의 안전성 설정

```
ALTER DATABASE [AdventureWorks] SET SAFETY FULL;
```

이 시점에서 미러링이 자동으로 시작되며 주 서버와 미러 서버는 동기화됩니다.

TIMEOUT 매개 변수를 사용하여 파트너 중단을 결정하는 시간 초과 값을 ALTER DATABASE로 조정할 수 있습니다. 예를 들어, TIMEOUT 값을 20초(기본값은 10)로 변경하려면 주 서버에서 다음 명령을 실행합니다.

-- Issue from the principal server

```
ALTER DATABASE [AdventureWorks] SET PARTNER TIMEOUT 20;
```

마지막으로 주 서버에서 ALTER DATABASE에 REDO\_QUEUE 옵션을 실행하여 미러에서 재실행 큐의 크기를 조정할 수 있습니다. 다음 쿼리는 미러에서 재실행 큐를 100MB로 설정합니다.

-- Issue from the principal server

```
ALTER DATABASE [AdventureWorks] SET PARTNER REDO_QUEUE 100MB;
```

파트너를 지정했으면 미러링이 즉시 시작됩니다.

## Database Mirroring 카탈로그 뷰

데이터베이스 미러링 세션은 파트너 서버와 감시 서버 사이에 구성된 관계로 구성됩니다. 각각의 참가하는 서버는 세션에 대한 일부 메타데이터와 데이터베이스의 현재 상태를 유지합니다. sys.database\_mirroring 카탈로그 뷰를 쿼리하여 주 서버나 미러 서버에서 세션을 검사할 수 있습니다. 다른 카탈로그 뷰를 사용하여 감시 서버 메타데이터가 반환됩니다.

sys.database\_mirroring\_witnesses. 두 카탈로그 뷰에서 사용할 수 있는 모든 열에 대한 자세한 내용은 SQL Server 온라인 설명서의 “sys.database\_mirroring” and “sys.database\_mirroring\_witnesses” 를 참조하십시오.

데이터베이스 미러링 세션의 작동 방식과 데이터베이스의 가능한 상태를 이해하는 좋은 방법은 카탈로그 뷰에서 데이터를 검사하는 것입니다. 안전성이 FULL이고 감시 서버가 있는 고가용성 세션부터 시작하겠습니다. 다음 쿼리는 주 또는 미러 서버에 대한 기본 데이터베이스 미러링 세션 정보에 대한 설명을 반환합니다.

```
SELECT
    DB_NAME(database_id) AS 'DatabaseName'
    , mirroring_role_desc
    , mirroring_safety_level_desc
    , mirroring_state_desc
    , mirroring_safety_sequence
    , mirroring_role_sequence
    , mirroring_partner_instance
    , mirroring_witness_name
    , mirroring_witness_state_desc
    , mirroring_failover_lsn
FROM sys.database_mirroring
WHERE mirroring_guid IS NOT NULL;
```

다음은 감시에서 실행되는 감시 서버에 대한 관련 설명 세션을 반환하는 유사 쿼리입니다.

```
SELECT
    Database_name
  , safety_level_desc
  , safety_sequence_number
  , role_sequence_number
  , is_suspended
  , is_suspended_sequence_number
  , principal_server_name
  , mirror_server_name
FROM sys.database_mirroring_witnesses;
```

이제 일반적인 데이터베이스 미러링 세션에서 두 쿼리의 출력을 비교할 수 있습니다. 서버 A에서 안전성이 FULL인 서버 B로 미러링을 설정했습니다. 다음 구성을 설정하는 예제 코드는 뒷부분에 있는 데이터베이스 미러링 구현의 “설정 및 보안”을 참조하십시오. 감시 서버는 서버 W이고 AdventureWorks 데이터베이스를 미러링합니다. 표 3은 두 파트너에 대한 위 쿼리의 가능한 출력을 보여줍니다.

**표 3. 두 파트너에 대한 예제 고가용성 세션에서 sys.database\_mirroring의 출력**

| 파트너<br>메타데이터<br>열            | 주 서버<br>값:<br>서버 A              | 미러 서버<br>값:<br>서버 B             |
|------------------------------|---------------------------------|---------------------------------|
| db_name(database_id)         | AdventureWorks                  | AdventureWorks                  |
| mirroring_role_desc          | PRINCIPAL                       | MIRROR                          |
| mirroring_safety_level_desc  | FULL                            | FULL                            |
| mirroring_state_desc         | SYNCHRONIZED                    | SYNCHRONIZED                    |
| mirroring_safety_sequence    | 1                               | 1                               |
| mirroring_role_sequence      | 1                               | 1                               |
| mirroring_partner_instance   | TCP://B.corp.mycompany.com:5022 | TCP://A.corp.mycompany.com:5022 |
| mirroring_witness_name       | TCP://W.corp.mycompany.com:5022 | TCP://W.corp.mycompany.com:5022 |
| mirroring_witness_state_desc | CONNECTED                       | CONNECTED                       |
| mirroring_failover_lsn       | 95000000007300001               | 95000000007300001               |

데이터베이스 미러링 세션의 각 파트너는 파트너의 관점에서 모두 같은 메타데이터를 유지합니다. 각 파트너는 자체 데이터베이스 이름, 전체 세션의 안전성 설정, 데이터베이스의 미러링 상태 및 두 시퀀스 카운터를 유지합니다.

- `mirroring_safety_sequence` 카운터는 두 파트너에서 유지되며 안전성 설정이 변경될 때마다 증분됩니다.
- `mirroring_role_sequence` 카운터는 두 파트너에서 유지되며 장애 조치가 발생할 때마다 증분됩니다.

두 파트너 데이터베이스 상태와 감시 상태는 각 파트너 서버에서 유지합니다.

- `mirroring_state_desc`는 파트너 데이터베이스가 세션 중에 있는 상태를 보여줍니다.
- `mirroring_witness_state_desc`는 세션에서 감시의 상태를 보여줍니다.

마지막으로 각 파트너에는 `mirroring_failover_lsn`이 포함되어 있습니다. LSN은 로그 순서 번호입니다. 이 번호는 각 트랜잭션 로그 레코드를 고유하게 식별합니다. 밀링 파트너는 세션에 대해 저장된 로그 레코드의 마지막 집합에서 가장 높은 LSN +1을 저장합니다. 위의 표에서는 주 데이터베이스의 작업량이 적었기 때문에 미러링 장애 조치 lsn은 주 서버와 감시 서버에서 동일했습니다. 많은 데이터를 전송할 때는 미러는 다소 뒤에서 실행되기 때문에 주 서버의 값이 미러 서버의 값보다 앞선 것을 알 수 있습니다.

이제 감시 서버에서 발견된 메타데이터를 비교해 봅시다. 다음 표는 감시 서버 메타데이터의 유사한 정보를 보여줍니다.

**표 4. 파트너 메타데이터와 관련된 감시 서버에서 sys.database\_mirroring\_witnesses의 출력**

| Witness<br>메타데이터<br>열        | Witness<br>값:                   | 해당 파트너<br>메타데이터<br>열        |
|------------------------------|---------------------------------|-----------------------------|
| database_name                | AdventureWorks                  | db_name(database_id)        |
| safety_level_desc            | FULL                            | mirroring_safety_level_desc |
| safety_sequence_number       | 1                               | mirroring_safety_sequence   |
| role_sequence_number         | 1                               | mirroring_role_sequence     |
| is_suspended                 | 0                               |                             |
| is_suspended_sequence_number | 1                               |                             |
| principal_server_name        | TCP://A.corp.mycompany.com:5022 |                             |
| mirror_server_name           | TCP://B.corp.mycompany.com:5022 |                             |

감시 서버의 메타데이터에는 이름은 약간 다르지만 안전성 설명, 안전성 시퀀스 번호 및 역할 시퀀스 번호가 포함되어 있습니다. 감시 서버는 세션의 일시 중단 여부에 대한 데이터와 주 서버 및 미러 서버 이름을 유지합니다. 감시 서버 카탈로그 뷰는 미러링 장애 조치 lsn을 보고하지 않으며 데이터베이스 상태를 유지하지 않습니다.

데이터베이스 미러링에 필요한 모든 메타데이터(특히, 미러링 장애 조치 lsn 및 파트너 서버 이름)를 미러링 파트너가 유지합니다. 감시 서버는 고가용성 모드에서 특히 세션에서 역할 변경 수를 추적하는 역할 시퀀스 번호 등 감시 역할에 필요한 데이터만 유지합니다. 이 카운터는 주 서버가 역할을 변경할 때를 결정하는 데 사용됩니다. 다음 절에서 시나리오를 배울 것입니다.

## Database Mirroring 상태 및 전환

각 서버에 대한 데이터베이스 상태는 데이터베이스 미러링 세션 동안 유지되며, 각 파트너 서버에 기록되고 sys.database\_mirroring 카탈로그 뷰에 의해 보고됩니다. mirroring\_state 열은 많은 상태를 반환하고 mirroring\_state\_desc 열은 상태에 대한 설명 이름을 반환합니다. 데이터베이스 상태 번호와 설명 이름의 전체 목록은 SQL Server 온라인 설명서의 "sys.database\_mirroring"을 참조하십시오. 감시에 대한 상태 정보도 동일한 카탈로그 뷰에서 보고됩니다.

각 데이터베이스에 대해 보고된 상태 이외에 데이터베이스 미러링에 관련된 서버와 데이터베이스를 설명하는 데 유용한 세 가지 구문이 있습니다.

1. 주 서버의 데이터는 트랜잭션을 처리 중이지만 로그 데이터가 미러로 보내지지 않을 때 표시됩니다.
2. 주 서버가 데이터베이스에 대한 사용자 연결과 트랜잭션 처리를 허용하지 않을 때 데이터베이스를 서비스할 수 없습니다.
3. 데이터베이스 미러링 세션에서 다른 서버에 연결할 수 없고 다른 서버가 데이터베이스에 연결할 수 없을 때 서버는 격리됩니다.

주 데이터베이스가 표시되면 사용자 연결과 트랜잭션 처리를 위해 활성화됩니다. 그러나 로그 레코드가 미러 데이터베이스에 보내지지 않고 주 서버에 오류가 발생하는 경우 미러는 표시된 상태로 들어온 지점에서 주 서버의 트랜잭션이 없습니다. 또한 운영 서버의 트랜잭션 로그가 잘리지 않기 때문에 로그 파일의 크기가 무한대로 증가하게 됩니다.

안전성이 FULL이면 주 서버가 다른 서버와 쿼럼을 구성할 수 없는 경우 데이터베이스 서비스가 중지됩니다. 주 데이터베이스에서 사용자 연결과 트랜잭션이 허용되지 않고 모든 현재 사용자의 연결이 끊어집니다. 다시 쿼럼을 구성할 수 있게 되면 데이터베이스 서비스로 돌아갑니다.

서버는 작동할 수 있지만 서버와 데이터베이스 미러링 세션에 있는 다른 두 서버 사이의 통신 회선이 다운됩니다. 이 경우 서버가 격리되었다고 합니다. 안전성이 FULL일 때 주 서버가 격리되면 세션에 쿼럼을 구성할 수 있는 서버가 없기 때문에 데이터베이스를 더 이상 서비스할 수 없게 됩니다.

이제 주 서버와 데이터베이스에서 시작하여 데이터베이스 미러링 상태에 집중해 보도록 합시다.

## 주 서버 데이터베이스 상태

안전성이 FULL이면 주 데이터베이스의 정상 작동 상태는 SYNCHRONIZED 상태입니다. 안전성이 OFF이면 주 데이터베이스의 정상 작동 상태는 SYNCHRONIZING 상태입니다.

- 안전성이 FULL이면 주 데이터베이스는 항상 SYNCHRONIZING 상태에서 시작하고 주 서버와 미러 서버 트랜잭션 로그가 동기화되면 SYNCHRONIZED 상태로 전환됩니다.
- 안전성이 FULL 이고 주 서버가 감시 서버에서 연결이 끊어졌지만 여전히 트랜잭션을 처리할 수 있는 경우 데이터베이스가 표시됩니다.
- 안전성이 FULL 이고 주 서버가 다른 서버와 쿼럼을 구성할 수 없는 경우 데이터베이스를 서비스할 수 없습니다. 사용자가 연결할 수 없고 트랜잭션을 처리하지 않습니다.

다음 표는 주 데이터베이스가 있을 수 있는 상태 및 다른 상태로 전환될 수 있는 일부 이벤트를 보여줍니다.

표 5. 안전성이 FULL이고 안전성이 OFF인 주 데이터베이스 상태

| 안전성  | 주 서버 초기 상태    | 이벤트              | 결과 상태        | 쿼럼          | 표시 | db 를 서비스 가능 |
|------|---------------|------------------|--------------|-------------|----|-------------|
| FULL | SYNCHRONIZING | 동기화 발생           | SYNCHRONIZED | Y           | N  | Y           |
| FULL | SYNCHRONIZED  | 세션 일시 중지         | SUSPENDED    | Y           | Y  | Y           |
| FULL | SYNCHRONIZED  | 미러 서버상의 다시 실행 오류 | SUSPENDED    | Y, 감시 서버 있음 | Y  | Y           |
|      |               |                  |              | N, 감시 서버 없음 | -  | N           |
| FULL | SYNCHRONIZED  | 미러 서버 사용 불가능     | DISCONNECTED | Y, 감시 서버 있음 | Y  | Y           |
|      |               |                  |              | N, 감시 서버 없음 | -  | N           |
| OFF  | SYNCHRONIZING | 세션 일시 중지         | SUSPENDED    | -           | Y  | Y           |
| OFF  | SYNCHRONIZING | 미러 서버상의 다시 실행 오류 | SUSPENDED    | -           | Y  | Y           |
| OFF  | SYNCHRONIZING | 미러 서버 사용 불가능     | DISCONNECTED | -           | Y  | Y           |

안전성이 FULL이면 먼저 주 서버는 SYNCHRONIZING 상태가 되고 미러와 동기화되는 즉시 두 파트너 모두 SYNCHRONIZED 상태가 됩니다. 안전성이 OFF이면 파트너 데이터베이스는 SYNCHRONIZING 상태로 시작하고 정상적인 미러링 동안 상태를 유지합니다.

두 안전성 설정 모두의 경우 세션이 일시 중단되거나 미러에 재실행 오류가 있는 경우 주 서버는 SUSPENDED 상태가 됩니다. 미러를 사용할 수 없게 되는 경우 주 서버는 DISCONNECTED 상태가 됩니다.

DISCONNECTED 및 SUSPENDED 상태에서:

- 안전성이 FULL일 때 주 서버가 감시 또는 미러 서버와 쿼럼을 구성할 수 있는 경우 주 데이터베이스는 표시된 것으로 간주됩니다. 즉, 사용자 연결과 트랜잭션 처리를 위해 주 데이터베이스가 활성화됩니다. 그러나 로그 레코드가 미러 데이터베이스에 보내지지 않고 주 서버에 오류가 발생하는 경우 미러는 해당 상태로 들어온 지점에서 주 서버의 트랜잭션이 없습니다. 또한 운영 서버의 트랜잭션 로그가 잘리지 않기 때문에 로그 파일의 크기가 무한대로 증가하게 됩니다.
- 안전성이 FULL이고 주 서버가 다른 서버와 쿼럼을 구성할 수 없는 경우 데이터베이스를 서비스할 수 없습니다. 모든 사용자의 연결은 끊어지고 새 트랜잭션이 처리되지 않습니다.
- 안전성이 OFF이면 트랜잭션 로그 레코드가 미러로 보내지지 않기 때문에 주 데이터베이스는 표시된 것으로 간주됩니다.

**주:** Management Studio의 Object Explorer는 Server 트리 그래프에서 데이터베이스 이름 옆에 주 데이터베이스 상태를 보고합니다. 주 서버의 SYNCHRONIZED 상태를 'Principal, Synchronizing' 로, DISCONNECTED 상태를 'Principal, Disconnected' 로 보고합니다.

### 미러 서버 데이터베이스 상태

미러 데이터베이스는 주 데이터베이스와 같은 데이터베이스 상태를 가지지만 미러는 항상 nonrecovered 상태에 있기 때문에 미러 역할에 있는 동안에는 데이터베이스를 서비스하지 못합니다. 다음 표는 미러 서버와 데이터베이스의 데이터베이스 상태를 변경시킬 수 있는 가장 일반적인 이벤트를 보여줍니다.

**표 6. 안전성이 FULL 및 OFF인 미러 데이터베이스 미러링 상태**

| 안전성  | 미러 서버 상태      | 이벤트                            | 결과 상태        |
|------|---------------|--------------------------------|--------------|
| FULL | SYNCHRONIZING | 동기화 발생                         | SYNCHRONIZED |
| FULL | SYNCHRONIZED  | 세션 일시 중지                       | SUSPENDED    |
| FULL | SYNCHRONIZED  | 미러 서버상의 다시 실행 오류               | SUSPENDED    |
| FULL | SYNCHRONIZED  | Principal database unavailable | DISCONNECTED |
| OFF  | SYNCHRONIZING | 세션 일시 중지                       | SUSPENDED    |
| OFF  | SYNCHRONIZING | 미러 서버상의 다시 실행 오류               | SUSPENDED    |

주 서버와 마찬가지로 Management Studio의 Object Explorer는 Server 트리의 데이터베이스 이름 옆에 일부 미러 데이터베이스 상태를 보고합니다. 미러 서버의 SYNCHRONIZED 상태를 'Mirror, Synchronizing' 로, DISCONNECTED 상태를 'Mirror, Disconnected' 로 보고합니다.

## 감시 서버 상태

감시 서버에는 sys.database\_mirroring 카탈로그 뷰에 CONNECTED, DISCONNECTED 및 UNKNOWN의 세 가지 상태가 있습니다.

표 7. 파트너 서버에 기록된 감시 서버 상태

| 감시 서버 상태  | 이벤트                 | 결과 상태        |
|-----------|---------------------|--------------|
| CONNECTED | 감시 서버 사용 불가능        | DISCONNECTED |
|           | 주 서버가 미러를 초기화할 수 없음 | UNKNOWN      |

감시 서버의 상태가 실제로 파트너 서버에 기록되고 감시 서버에는 기록되지 않기 때문에 이러한 상태는 파트너의 유리한 지점에서 설정됩니다. 따라서 감시 서버의 상태가 DISCONNECTED이면 파트너가 감시 서버에서 연결이 끊어진 것을 의미합니다. 데이터베이스 미러링이 시작될 때 미러가 주 서버와 초기화할 수 없는 경우 감시 서버 상태는 UNKNOWN이 됩니다.

## 트랜잭션 로그 레코드 전송

주 및 미러 SQL Server가 메시지와 로그 레코드 블록을 전송하는 데 사용하는 작업 순서는 안전성 설정에 따라 다릅니다. 먼저 동기 전송을 검사한 다음 비동기 전송을 검사합니다.

SQL Server가 트랜잭션 로그에 트랜잭션 이벤트를 기록하면 로그 레코드는 디스크에 기록하기 전에 로그 버퍼에 잠시 저장됩니다. 데이터베이스 미러링에서 로그 버퍼를 디스크에 기록(저장)할 때마다 주 서버는 미러 서버에 동일한 로그 레코드 블록을 보냅니다.

1. 안전성이 FULL일 때 주 SQL Server가 로그 레코드 블록을 저장하면 동일한 블록을 미러로 보내고 로컬 I/O를 로그로, 원격 미러 서버의 I/O를 로그에 처리합니다. 트랜잭션이 완료된 것으로 간주되려면 I/O(저장)가 완료되었다는 로컬 I/O(저장)와 미러의 응답을 주 서버가 기다리기 때문에 전송을 동기화라고 부릅니다.

주 서버나 미러 서버가 로그 버퍼를 저장할 때마다 메타데이터에 있는 버퍼의 가장 높은 로그 시퀀스 번호(LSN) + 1을 mirroring\_failover\_lsn으로 기록합니다. mirroring\_failover\_lsn은 트랜잭션 로그에서 최신의 보장되는 지점을 협상하는 데 사용되므로 두 파트너 데이터베이스는 초기에 동기화하고 장애 조치 후에 동기화할 수 있습니다.

미러링 장애 조치 lsn은 주 서버가 미러에 로그 레코드를 보낼 때 항상 주 서버에 미리 있게 됩니다. 로그 레코드를 저장할 때 미러는 mirroring\_failover\_lsn을 기록하고 주 서버에 응답하지만 주 서버가 미러로부터 승인을 받으면 주 서버는 새로운 로그 레코드 집합을 저장할 수 있습니다.

표 8은 안전성이 FULL인 주 서버와 미러 사이의 예제 이벤트 시퀀스를 보여줍니다.

표 8. 이벤트의 안전성 FULL(동기 전송) 예제 시퀀스

| 서버 A                                                                                                                                                                                     | 서버 B                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| 주 서버, Synchronized                                                                                                                                                                       | 미러 서버, Synchronized                                                                          |
| 여러 명령문의 트랜잭션이 데이터 변경 내용을 포함하기 시작합니다.                                                                                                                                                     |                                                                                              |
| 주 데이터베이스의 트랜잭션 로그 레코드가 트랜잭션 로그 버퍼에 삽입됩니다.                                                                                                                                                |                                                                                              |
| 트랜잭션 로그 버퍼는 디스크(하드엔디드)에 작성되고 로그 레코드 블록은 미러 서버로 전송됩니다.<br>및<br>주 서버는 해당 로그 블록의 mirroring_failover_lsn을 기록합니다.<br>및<br>주 서버는 미러 서버의 확인을 기다립니다                                              |                                                                                              |
|                                                                                                                                                                                          | 미러 서버는 로그 레코드 블록을 트랜잭션 로그 버퍼로 수신합니다.                                                         |
|                                                                                                                                                                                          | 미러 서버는 로그 버퍼를 디스크에 작성하고 mirroring_failover_lsn을 기록하며<br>및<br>로그 블록이 보호되고 있다는 것을 운영 서버에 알립니다. |
| 운영 서버는 로그 레코드 블록이 미러 서버에 의해 디스크에 작성되었다는 알림 메시지를 받습니다.                                                                                                                                    | 미러 서버는 REDO 대기열에서 계속해서 트랜잭션 로그 레코드를 재실행합니다.                                                  |
| 트랜잭션에 대한 COMMIT이 트랜잭션 로그 버퍼에 입력됩니다.<br>트랜잭션 로그 버퍼는 디스크(하드엔디드)에 작성되고 COMMIT이 들어 있는 로그 레코드 블록은 미러 서버로 전송됩니다.<br>주 서버는 해당 블록의 mirroring_failover_lsn을 기록합니다.<br>및<br>주 서버는 미러 서버의 확인을 기다립니다 |                                                                                              |
|                                                                                                                                                                                          | 미러 서버는 로그 레코드 블록을 트랜잭션 로그 버퍼로 수신합니다.                                                         |
|                                                                                                                                                                                          | 미러 서버는 로그 버퍼를 디스크에 작성하고 mirroring_failover_lsn을 기록하며<br>및<br>로그 블록이 보호되고 있다는 것을 운영 서버에 알립니다. |
| 운영 서버는 로그 레코드 블록이 미러 서버에 의해 디스크에 작성되었다는 알림 메시지를 받습니다.<br>및<br>트랜잭션은 커밋된 것으로 간주됩니다.                                                                                                       | 미러 서버는 REDO 대기열에서 COMMIT을 포함한 트랜잭션 로그 레코드를 재생하여 데이터 페이지가 변경됩니다.                              |
| 새로운 트랜잭션은 주 서버의 로그 버퍼에 기록됩니다.                                                                                                                                                            |                                                                                              |

앞의 시퀀스에서 중요한 점은 안전성이 FULL이고, 주 서버가 로그 버퍼를 저장하고 동시에 버퍼에서 미러 서버로 로그 레코드의 복사본을 보낸다는 것입니다. 그런 다음 트랜잭션이 완료되었다고 간주하기 전에 자체 I/O 및 미러 서버의 I/O가 완료되기를 기다립니다. 주 서버가 미러 서버에서 응답을 받으면 주 서버는 다음 저장으로 진행할 수 있습니다.

안전성이 FULL일 때 주 서버와 미러 서버 사이의 밀접한 협력에도 불구하고 데이터베이스 미러링은 배포된 트랜잭션이 아니고 두 단계 커밋을 사용하지 않습니다.

- 데이터베이스 미러링에서는 두 서버에서 한 트랜잭션이 배포되는 것이 아니라 두 트랜잭션이 재생됩니다.
- 데이터베이스 미러링은 배포된 트랜잭션에서 파트너 서버를 리소스 관리자로 사용하지 않습니다.
- 데이터베이스 미러링 트랜잭션은 준비와 커밋 단계를 거치지 않습니다.
- 가장 중요한 것은 배포된 트랜잭션과 달리 미러 서버에서 커밋하지 못하면 주 서버에서 트랜잭션이 롤백되지 않습니다.

2. 안전성이 OFF이면 주 서버가 미러 서버의 승인을 기다리지 않으므로 주 서버에서 커밋된 트랜잭션 수는 표 9에 표시된 대로 미러에 앞서 얻을 수 있습니다.

**표 9. 이벤트의 안전성 OFF(비동기 전송) 예제 시퀀스**

| 서버 A                                                                                                       | 서버 B                                                                                                          |
|------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| 주 서버, Synchronizing                                                                                        | 미러 서버, Synchronizing                                                                                          |
| 여러 명령문의 트랜잭션이 데이터 변경 내용을 포함하기 시작합니다.                                                                       |                                                                                                               |
| 데이터 수정을 위한 트랜잭션 로그 레코드가 트랜잭션 로그 버퍼에 기록됩니다.                                                                 |                                                                                                               |
| 트랜잭션 로그 버퍼는 디스크(하드엔디드)에 플러시되고 로그 레코드 블록은 미러 서버로 전송됩니다.<br>및<br>주 서버는 해당 블록의 mirroring_failover_lsn을 기록합니다. |                                                                                                               |
| 트랜잭션의 COMMIT는 다른 트랜잭션 작업과 함께 트랜잭션 로그 버퍼에 입력됩니다.                                                            | 미러 서버는 로그 레코드 블록을 트랜잭션 로그 버퍼로 수신합니다.                                                                          |
| 트랜잭션 로그 버퍼는 디스크에 저장됩니다.<br>및<br>COMMIT이 들어 있는 로그 블록은 미러 서버로 전송됩니다.                                         | 미러 서버는 로그 버퍼를 디스크에 작성하고 mirroring_failover_lsn을 기록하며<br>및<br>로그 블록이 보호되고 있다는 것을 운영 서버에 알립니다.                  |
| 트랜잭션은 커밋된 것으로 간주됩니다.                                                                                       | 미러 서버는 REDO 대기열에서 계속해서 트랜잭션 로그 레코드를 재실행합니다.                                                                   |
|                                                                                                            | 미러 서버는 로그 레코드 블록을 트랜잭션 로그 버퍼로 수신합니다.                                                                          |
|                                                                                                            | 미러 서버는 디스크에 기록하여 로그 블록을 저장합니다.<br>및<br>mirroring_failover_lsn을 기록합니다.<br>및<br>로그 블록이 보호되고 있다는 것을 운영 서버에 알립니다. |

## 데이터베이스 미러링 역할 변경

데이터베이스 미러링 장애 조치는 데이터베이스 미러링 서버 또는 응용 프로그램의 관점에서 문제점으로 간주될 수 있습니다. 데이터베이스 미러링 서버의 관점에서 장애 조치는 미러 서버를 주 서버로 변환하고 새로 복구한 데이터베이스를 세션의 주 데이터베이스로 사용합니다. 장애 조치는 자동, 수동 또는 강제가 될 수 있습니다.

- **자동** - 고가용성 모드에서만 발생합니다(안전성은 FULL이고 감시 서버가 세션의 일부입니다).
- **수동** - 고가용성 및 높은 수준의 보호 작동 모드(안전성이 FULL)에서 발생하며 파트너 데이터베이스 둘 모두 동기화됩니다.
- **강제 서비스(데이터 손실 허용)** - 고성능(안전성이 OFF) 모드에서 주로 사용되며 미러 데이터베이스를 즉시 수동으로 복구합니다.

안전성이 FULL이면 서버의 역할을 반대로 바꾸는 가장 좋은 방법은 강제 서비스가 아닌 수동 장애 조치를 사용하는 것입니다.

### 자동 장애 조치

자동 장애 조치는 고가용성 모드(감시 서버의 안전성이 FULL)에서 데이터베이스 미러링의 기능입니다. 대부분의 경우 SQL Server는 몇 초 내에 자동 데이터베이스 미러링 장애 조치를 수행할 수 있습니다. 데이터베이스 미러링 세션에 참여하는 SQL Server가 모두 서로의 존재를 테스트하기 때문에 부분적으로 이것을 수행할 수 있습니다. 이 프로세스는 일반적인 IP 주소 ping보다 훨씬 많이 관련되지만 'ping'이라고 합니다. 미러 서버와 감시 서버는 실제 서버가 있는지, SQL Server 인스턴스가 있는지, 주 데이터베이스의 가용성 확인을 위해 주 서버에 연결합니다. 마찬가지로, 실제 서버의 가용성, SQL Server 인스턴스 및 미러 데이터베이스의 복구 상태 확인을 위해 주 서버와 감시 서버는 미러 서버에 핑(ping)합니다.

데이터베이스 미러링 세션이 안전성이 FULL이고 감시 서버를 사용하도록 설정되었다고 가정합니다. 미러 서버인 서버 B는 주 서버 A를 사용할 수 없다는 것을 ping을 통해 발견합니다. 서버 B는 감시 서버와 통신하며 감시 서버가 서버 A를 볼 수 없다는 확인을 받습니다. 그러면 서버 B는 감시 서버와 쿼럼을 구성하며 주 서버로 수준이 올라갑니다. 데이터베이스는 연결이 끊어진 상태이고 새로운 주 데이터베이스의 트랜잭션 로그를 잘라낼 수는 없지만 데이터베이스를 복구하고 이제 주 서버 역할을 갖게 되었음을 감시 서버에 알립니다.

서버 B의 새로운 주 데이터베이스는 트랜잭션 로그 작업을 계속 재생하지만 지속적으로 재실행 상태에 있었으며 대부분의 경우 수행할 일이 거의 남아 있지 않습니다. 모든 SQL Server 버전의 데이터베이스 미러링에서 새로운 주 데이터베이스는 재실행 상태를 완료하는 즉시 사용할 수 있게 됩니다. 데이터베이스가 실행 취소 상태가 되면 사용자 연결을 사용할 수 있게 됩니다. 남아 있는 실행 취소 단계는 훨씬 오래 걸릴 수 있지만 재실행 상태를 완료하는 데 보통 몇 초 정도 걸립니다. 데이터베이스 미러링에서 새로운 주 데이터베이스는 재실행 프로세스가 완료되는 즉시 사용자 연결을 서비스하는 데 사용할 수 있게 됩니다. 새로운 주 서버 B의 데이터베이스는 DISCONNECTED 상태에 있고 표시되지만 재실행 단계가 완료되는 즉시 데이터베이스를 서비스할 수 있습니다.

대개 데이터베이스 미러링 자동 장애 조치에 걸리는 시간보다 기존의 주 서버에서 새로운 주 서버로 전체 응용 프로그램을 리디렉션하는 데 더 많은 시간이 걸립니다. 응용 프로그램은 연결을 감지하고 재시도해야 하며, 이는 프로세스에 약간의 시간을 추가할 수 있습니다. 또한 SQL Server 인증을 사용한 새로운 로그인인 서버에 추가된 경우 시스템 저장 프로시저 sp\_change\_users\_login 을 사용하여 이러한 로그인을 새로운 주 서버에 매핑해야 할 수 있습니다. SQL 에이전트 작업 같이 이전 서버의 중요한 개체가 새로운 주 서버에 복사되지 않은 경우 전체 응용 프로그램 장애 조치는 지연될 수도 있습니다. 자세한 내용은 뒷부분의 구현 절에서 "장애 조치를 위해 미러 서버 준비"를 참조하십시오.



이제 기존의 주 서버가 온라인이 되는 것을 가정합니다. 수동 장애 조치에서처럼 역할이 반대이거나 기존의 주 서버가 아주 빨리 복구되는 자동 장애 조치인 경우 두 서버 간에 발생해야 하는 협상 프로세스가 있습니다. 미러링을 다시 시작하기 전에 두 파트너 서버는 서로 동기화하는 방법을 찾아야 합니다. 미러링 장애 조치 lsn은 이 프로세스에서 중요한 역할을 수행합니다.

서버 A(새로운 미러)는 뒤에 있지만 어느 정도인지는 확실하지 않습니다. 서버 A는 서버 B(새로운 주 서버)에 서버 B로부터 받은 마지막 미러링 장애 조치를 보고합니다. 반면에 서버 B에는 미러링 장애 조치 lsn을 최신 상태로 만든 커밋된 작업이 있습니다. 두 서버는 서버 B에 올바른 장애 조치 lsn이 있고 서버 A를 추적해야 한다는 것에 동의합니다. 서버 B는 동기화하기 위해 재생할 수 있는 충분한 수의 트랜잭션 레코드를 서버 A로 전송합니다.

### 수동 장애 조치

수동 장애 조치는 두 파트너 서버가 순서대로 오류 없이 자신의 역할을 바꾸는 방법입니다. 안전성이 FULL로 설정되고 주 데이터베이스와 미러 데이터베이스가 SYNCHRONIZED 상태에 있어야 합니다.

주 서버에서 ALTER DATABASE 명령을 호출하여 수동 장애 조치를 수행합니다.

```
ALTER DATABASE AdventureWorks SET PARTNER FAILOVER;
```

또는 Management Studio의 Database Properties/Mirroring 대화 상자에서 Failover 단추를 클릭합니다. 수동 장애 조치를 수행하면 현재 사용자의 연결이 끊어지고 기존의 주 데이터베이스에서 완료되지 않은 트랜잭션이 롤백됩니다. 재실행 대기열에서 모든 완료된 트랜잭션을 완료하고 실행 취소 단계에서 완료되지 않은 트랜잭션을 롤백하여 미러 데이터베이스를 복구합니다. 기존의 미러는 주 역할이 할당되고 기존의 주 데이터베이스는 미러의 새 역할을 맡습니다. 두 서버는 미러링 장애 조치 lsn을 기반으로 미러링의 새로운 시작점을 협상하고 반대의 역할을 사용하여 처리됩니다.

미러링을 초기화하기 전에 먼저 미러 서버를 업그레이드하는 경우 서버 운영 체제 또는 SQL Server 인스턴스에 '롤링 업그레이드'를 수행하는 방법으로 수동 장애 조치를 사용할 수 있습니다. 자세한 내용은 SQL Server 온라인 설명서의 'Manual Failover'를 참조하십시오.

### 강제 서비스(Forced Service)

ALTER DATABASE 명령을 호출하여 미러에서 강제 서비스를 수행할 수 있습니다.

```
ALTER DATABASE AdventureWorks SET PARTNER FORCE_SERVICE_ALLOW_DATA_LOSS;
```

일반적으로 이것은 안전성이 OFF이고 주 서버가 더 이상 작동하지 않는 경우에만 유용합니다. 또한 안전성이 FULL일 때 이 명령을 사용할 수도 있지만 복구된 미러 서버가 쿼럼을 구성할 수 없는 경우 데이터베이스를 서비스할 수 없습니다. 따라서 안전성이 OFF(고성능 모드)인 상태에서만 이 명령을 사용하는 것이 좋습니다. 비동기 데이터 전송은 주 서버에서 트랜잭션이 커밋된 날짜까지 미러를 유지할 수 없기 때문에 일부 데이터가 손실될 수 있습니다.

## 데이터베이스 미러링 가용성 시나리오

이 절에서는 다음 두 가지 유형의 이벤트를 기반으로 데이터베이스 미러링 시나리오의 예상되는 가용성 결과를 알아 보겠습니다.

- 하나 이상의 서버 또는 데이터베이스가 다운된 경우
- 서버 사이에 하나 이상의 통신 회선이 끊어진 경우

파트너 데이터베이스 중 하나 또는 SQL Server 인스턴스 중 하나를 사용할 수 없게 될 때 서버 손실이 발생할 수 있습니다. 또는 서버 자체는 계속 작동하지만 데이터베이스 미러링 파트너 서버 사이의 통신 회선이 중단될 수 있습니다.

다음 시나리오는 한 구성 요소가 실패하고 이어서 다른 구성 요소가 실패하여 두 구성 요소가 동시에 실패한 것으로 간주됩니다. 예를 들어, 서버 A와 B가 동시에 실패한 것으로 나타나는 경우 미러링 시스템은 이 이벤트를 순서대로 처리합니다. 서버 A에 이어 서버 B 또는 이와 반대로 처리합니다.

**다음 규칙을 사용하여 사용 불가능 이벤트의 예상 결과를 확인할 수 있습니다.**

1. 안전성이 FULL이면 주 서버는 데이터베이스 서비스를 유지하기 위해 최소 하나의 다른 서버에 쿼럼이 필요합니다.

**주 서버가 쿼럼을 구성할 수 없는 경우 더 이상 데이터베이스를 서비스할 수 없습니다.**

2. 안전성이 FULL일 때 미러나 감시 서버가 주 서버를 볼 수 없는 경우 미러 서버는 감시 서버와 쿼럼을 구성할 수 있고 새로운 주 서버가 되도록 감시 서버와 쿼럼을 구성할 수 있습니다.

**이는 자동 장애 조치입니다.**

3. 안전성이 FULL이고 감시 서버와 쿼럼을 구성한 상태에서 주 서버가 수행되지만 미러에서는 연결이 끊긴 경우 주 서버 실패는 미러가 감시 서버와 쿼럼을 구성하고 새로운 주 서버의 역할을 맡는 것이 허용되지 않습니다.

**이렇게 하면 세션에 의해 작업이 손실되는 것이 방지됩니다.**

4. 안전성이 FULL일 때 실패한 주 서버가 다운 또는 격리된 후에 세션에 다시 참가하고 기존의 미러 서버가 감시 서버가 있는 쿼럼에서 이전에 새로운 주 서버의 역할을 맡은 경우 기존의 주 서버는 세션에서 새로운 미러의 역할을 맡게 됩니다.

장애 조치 이벤트 동안 미러와 감시 서버는 미러링 역할 시퀀스 카운터를 증가시켰습니다. 기존의 주 서버 미러링 역할 시퀀스 카운터가 다른 파트너 서버 및 감시 서버의 시퀀스 카운터보다 작기 때문에 이러한 두 서버는 기존의 주 서버가 세션에 다시 참가하기 전에 쿼럼을 구성했고 새로운 주 서버에서 작업이 발생했을 수 있습니다. 기존의 주 서버는 세션에서 미러 역할을 맡습니다.

5. 안전성이 FULL이고 세션에 감시 서버가 없거나, 감시 서버가 세션에서 제거된 경우 쿼럼이 없으면 파트너 서버 실패가 발생하므로 주 서버는 더 이상 주 데이터베이스 서비스를 유지할 수 없습니다.

쿼럼이 가능하지 않으므로 감시 서버가 안전성 FULL 세션의 일부가 아닐 경우 자동 장애 조치가 발생할 수 없습니다.

## 서버 손실 시의 고가용성 시나리오

고가용성 작동 모드에서 데이터베이스 미러링의 목적은 가능한 데이터베이스를 사용할 수 있도록 유지하는 것입니다. 이 모드에서 데이터베이스 미러링은 주 데이터베이스를 사용할 수 없게 될 경우 미리 데이터베이스를 신속하게 서비스하게 됩니다. 다음 시나리오에서는 세 개의 독립 서버가 있는 고가용성 구성으로 시작합니다.

다음 고가용성 시나리오에서 그림 1에서 볼 수 있듯이 서버 A는 주 서버, 서버 B는 미러 서버, 서버 W는 감시 서버로 시작합니다.

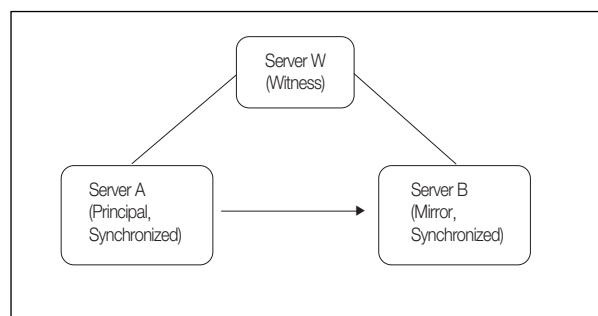


그림 1. 예제 데이터베이스 미러링 세션은 고가용성 작동 모드에서 시작합니다.

실제로 세 개의 모든 서버는 같은 사이트에 있고 로컬 네트워크를 통해 연결되거나 WAN을 통해 연결된 독립 사이트에 있을 수 있습니다. 서버 A와 서버 B는 역할을 변경할 수 있지만 서버 W는 항상 감시 서버를 유지합니다.

이제 서버 중 하나가 다운되는 경우 어떤 일이 발생하는지 생각해 봅시다.

### 시나리오 HASL1.1: 주 서버 손실

다음 시나리오는 주 서버가 고가용성 시나리오에서 손실될 때 어떤 일이 발생하는지 고려합니다. 그림 2는 다양한 역할 및 패트너 사이에서 어떻게 변하는지 보여줍니다.

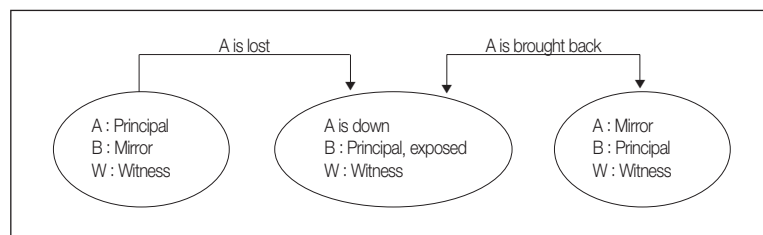


그림 2. 고가용성 모드에서 주 서버 A가 실패하는 경우 장애 조치가 발생

주 서버 A가 손실된 후에 미러 서버와 감시 서버가 쿼럼을 구성하고 자동 장애 조치가 발생합니다. 원본 주 서버를 다시 불러올 수 있는 경우 미러 역할을 맡게 됩니다.

주: 고가용성 모드에서 장애 조치가 발생하도록 하려면 많은 수준에서 실패가 발생할 수 있습니다. 실제 서버가 다운되거나, 주 서버의 SQL Server 인스턴스가 중지 또는 실패하거나, 서버의 주 데이터베이스를 사용하지 못할 수 있습니다. 다음 시나리오에서 주 서버가 손실되면 이러한 이벤트에 의해 발생할 수 있습니다.

서버 B와 W가 쿼럼을 구성할 수 있고 어느 서버도 서버 A를 볼 수 없기 때문에 서버 B가 새로운 주 서버로 자신의 수준을 올릴 수 있습니다. 그러나 미리 서버가 없으면 미러링 세션이 표시된 것으로 간주됩니다.

서버 A가 복원되면 새로운 주 서버가 되고 미러링 세션은 더 이상 표시되지 않습니다.exposed

단일 서버 실패는 거의 발생하지 않는 이벤트일 수 있습니다. 드물게 발생하는 이벤트라도 두 서버가 실패할 수 있습니다. 드문 경우라도 결과를 검사하는 것이 유용합니다.

두 서버가 동시에 또는 거의 동시에 실패할 수 있지만 데이터베이스 미러링 관점에서 결과는 마치 한 서버가 실패하고 이어서 다른 서버가 실패하는 것이 됩니다. 따라서 이러한 시나리오는 서버가 순차적으로 실패할 때만 발생하는 것으로 간주됩니다. 다음 두 시나리오는 주 서버 A 실패가 두 개의 대체 서버 실패에 이어 발생하는 것으로 간주합니다.

- 새로운 주 서버 B의 실패
- 감시 서버 W의 실패

#### 시나리오 HASL1.2: 주 서버 손실에 이어 새로운 주 서버 손실

앞의 시나리오에서 본 것처럼 고가용성 모드에서는 주 서버가 먼저 손실되는 경우 장애 조치가 발생합니다. 그림 3은 새로운 주 서버가 다음에 실패하는 경우 서버 복원에 관계 없이 새로운 주 서버(서버 B)가 주 서버 역할을 유지합니다.

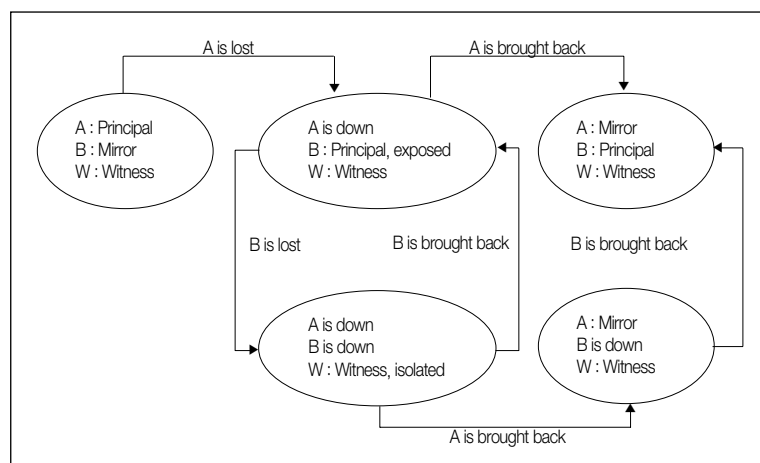


그림 3. 주 서버 손실에 이어 새로운 주 서버 손실로 인한 역할 변경

서버 A가 실패한 후에 서버 B가 새로운 주 서버가 되지만 데이터를 미리 서버로 전송할 수 없으므로 데이터베이스는 여전히 서비스할 수 있어도 주 서버는 표시됩니다. 서버 A 실패에 이어 서버 B가 실패하면 서버 B는 다운되기 때문에 미러링이 없습니다.

먼저 서버 A를 복구하는 경우 감시 서버가 새로운 쿼럼을 구성했기 때문에 감시 서버 W에서 mirroring\_role\_sequence 번호를 감지합니다. 서버 A는 미러의 역할을 맡고 서버 B가 복구되기를 기다립니다. 서버 B가 복구되자마자 서버 A로 미러링 프로세스를 시작합니다. 먼저 서버 B를 복구하는 경우에는 HASL1.1에 표시된 원래 시나리오로 돌아갑니다.

**주:** 서버 A가 손실된 다음 서버 B가 손실된 후에 서버 W가 손실된 경우 이러한 서버는 모두 다운되고 서버 A와 서버 B의 전환된 역할은 서버 복원 순서에 관계 없이 유지됩니다.

### 시나리오 HASL1.3. 주 서버 손실에 이어 감시 서버 손실

주 서버가 손실되면 장애 조치가 발생합니다. 그림 4에서와 같이 감시 서버가 다음에 실패할 수 있습니다.

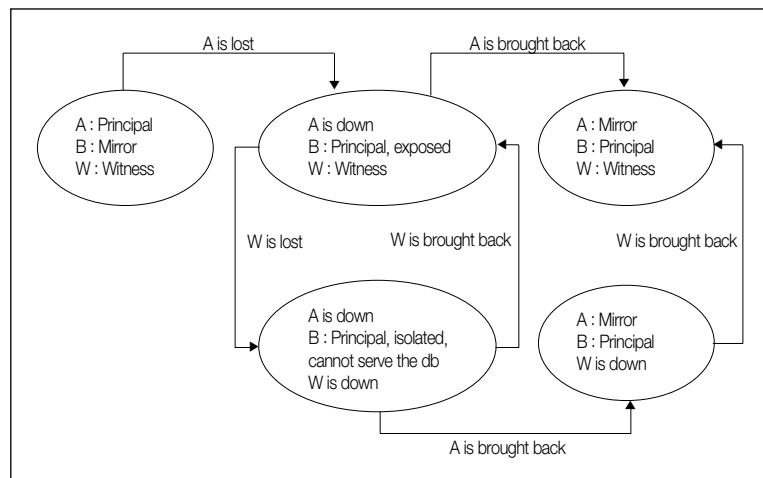


그림 4. 감시 서버 손실에 이어 초기 주 서버가 손실되면 새로운 주 서버가 데이터베이스를 서비스할 수 없습니다.

주 서버 A가 손실된 후에 감시 서버 W가 손실되면 새로운 주 서버 B는 여전히 주 서버이지만 격리되어 있어 쿼럼을 구성할 수 없고 데이터베이스를 서비스할 수 없습니다.

먼저 서버 A를 복구하는 경우 서버 B의 mirroring\_role\_sequence 번호는 장애 조치가 발생했기 때문에 서버 A의 번호보다 1이 커집니다. 이는 서버 A가 수행한 후에 서버 B가 이제 주 서버 역할을 가지는 것을 서버 A에 알립니다. 서버 A는 서버 B와 쿼럼을 구성하고 미러 서버가 되며 두 서버 모두 동기화됩니다. 서버 W가 복구될 때까지 자동 장애 조치는 발생할 수 없습니다.

**주:** 서버 A가 손실된 다음 서버 W가 손실된 후에 서버 B가 손실된 경우 서버 A와 서버 B의 새로운 역할은 서버 복원 순서에 관계 없이 유지됩니다.

### 시나리오 HASL2.1 미러 서버 손실

미러 서버가 먼저 손실된 경우 미러 서버로 데이터를 전송할 수 없기 때문에 주 서버가 노출된 것으로 간주됩니다. 그림 5는 미러 서버인 서버 B가 손실될 때의 역할을 보여줍니다.

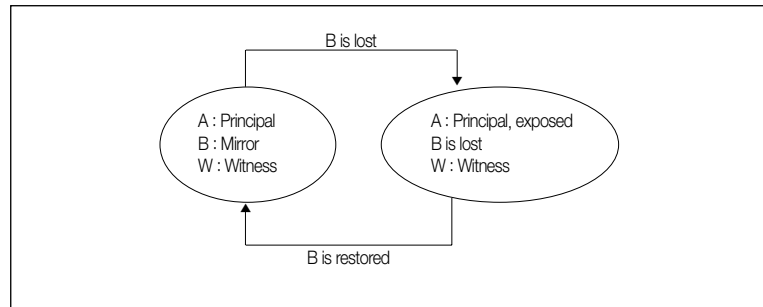


그림 5. 고가용성 모드에서 미러 서버 B가 실패하는 경우 장애 조치가 발생하지 않습니다.

자동 장애 조치는 발생하지 않으며 파트너는 역할을 교환하지 않습니다. 서버 B가 복원되면 세 서버 모두 원래 역할과 상태로 돌아갑니다.

다음 표는 미러 서버 B의 손실과 복원 동안 데이터베이스 상태와 쿼럼을 보여줍니다.

데이터는 중복 데이터베이스에 저장되지 않기 때문에 미러 서버 없이 세션이 노출됩니다.

서버 B를 복원할 수 있게 되면 미러 역할이 재개되며 두 파트너가 동기화되면 미러링 세션은 더 이상 노출된 것으로 간주되지 않습니다.

다음 두 시나리오는 미러 서버 B 실패에 이어 주 서버 A 또는 감시 서버 W가 실패할 때 발생하는 것으로 간주됩니다.

### 시나리오 HASL2.2: 미러 서버 손실에 이어 주 서버 손실

미러 서버에 이어 주 서버가 손실되면 파트너 서버의 역할은 변경되지 않습니다. 그림 6은 서버를 복구하는 다양한 경로를 취할 때 역할이 어떻게 변경되는지를 보여줍니다.

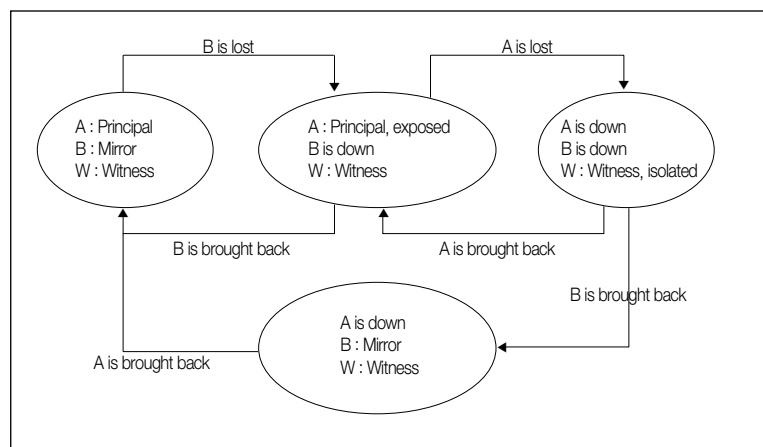


그림 6. 미러 서버에 이어 주 서버가 손실되는 경우 감시 서버는 격리됩니다.

서버 B에 이어 서버 A가 손실되면 상태는 다이어그램 오른쪽 위 모서리에 표시된 상태가 됩니다.

먼저 서버 B를 복구하는 경우 서버 A는 계속 주 서버이고 장애 조치가 발생하지 않는 감시 서버 W에서 발견됩니다. `mirroring_failover_lsn`은 증가하지 않습니다. 결과적으로 서버 B는 미러 서버를 유지합니다. 그런 다음 서버 W를 복구하면 세션이 원래 상태로 복원됩니다.

**주:** 서버 B에 이어 서버 A가 손실된 후에 서버 W가 손실되는 경우 어떤 순서로 서버를 복구하든 결과는 같습니다.

### 시나리오 HASL2.3: 미러 서버에 이어 감시 서버 손실

미러 서버 손실에 이어 감시 서버가 손실되면 주 서버는 격리되고 다른 서버와 쿼럼을 구성할 수 없습니다. 따라서 그림 7의 오른쪽 위 모서리에서 볼 수 있듯이 데이터베이스를 서비스하지 못하게 됩니다.

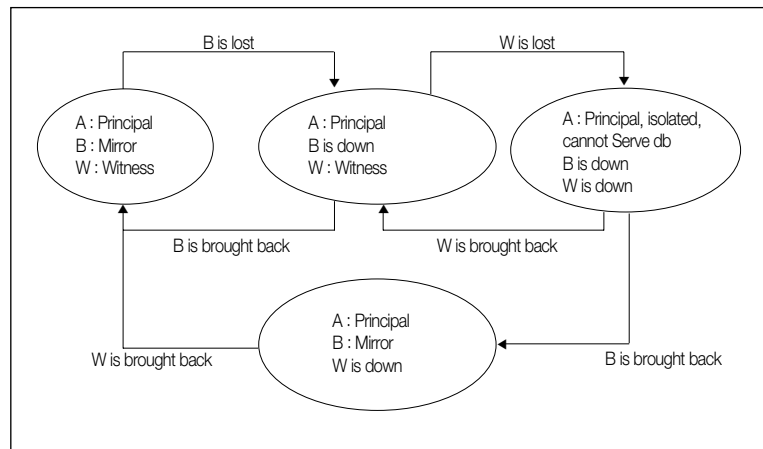


그림 7. 미러 서버 손실에 이어 감시 서버가 손실되면 주 서버가 데이터베이스를 서비스할 수 없게 됩니다.

미러 서버에 이어 감시 서버가 중단되면 주 서버 A는 주 서버 역할을 유지하지만 다른 서버와 쿼럼을 구성할 수 없고 안전성이 FULL이기 때문에 데이터베이스를 서비스할 수 없고 모든 사용자의 연결이 끊어집니다.

먼저 서버 B가 복원되면 감시 서버 없이는 자동 장애 조치가 가능하지 않지만 미러링은 계속됩니다.

먼저 서버 W가 복원되는 경우 시나리오는 그림 5와 같습니다.

**주:** 서버 B에 이어 서버 W가 손실된 후에 서버 A가 손실되는 경우 어떤 순서로 서버를 복구하든 최종 결과는 같습니다.

### 시나리오 HASL3.1: 감시 서버 손실

감시 서버가 실패하면 미러링은 계속되지만 자동 장애 조치는 가능하지 않습니다. 하나 이상의 서버 손실은 쿼럼이 없고 주 데이터베이스가 더 이상 데이터베이스를 서비스할 수 없음을 의미합니다.

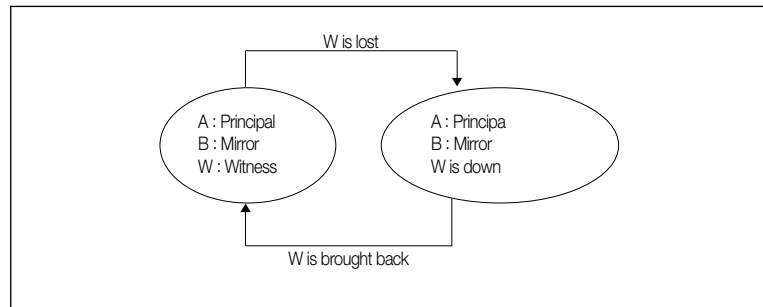


그림 8. 고가용성 모드에서 먼저 감시 서버 W가 실패하는 경우 미러링은 계속됩니다.

서버 W를 복원하면 파트너 서버 A와 서버 B는 원래 역할을 유지합니다.

다음 표는 감시 서버의 실패와 복원 동안 데이터베이스 상태와 쿼럼의 변경을 보여줍니다.

다음 두 시나리오는 감시 서버 W 실패에 이어 주 서버 A 또는 미러 서버 B가 실패할 때 발생하는 것으로 간주됩니다.

### 시나리오 HASL3.2: 감시 서버 손실에 이어 주 서버 손실

먼저 감시 서버가 실패하면 미러링은 계속되지만 자동 장애 조치는 가능하지 않습니다. 나머지 두 서버 중 하나가 실패하면 쿼럼은 가능하지 않고 나머지 서버는 격리됩니다.

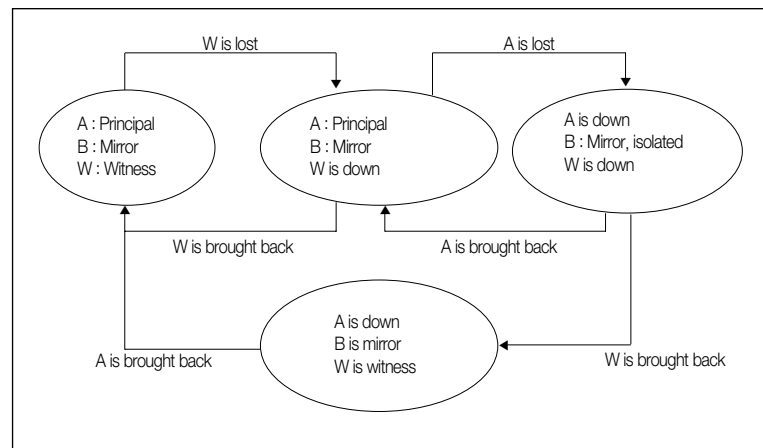


그림 9. 초기 감시 서버 손실에 이어 주 서버가 손실된 후에 파트너 역할은 동일하게 유지됩니다.

먼저 서버 W를 복구하는 경우 서버 B는 마지막 양호한 주 서버가 서버 A인 감시 서버에서 감지하므로 서버 B는 미러 서버를 유지합니다. 마지막으로 서버 A를 복구하면 주 서버 역할을 유지합니다.

**주:** 서버 W에 이어 서버 A가 손실된 후에 서버 B가 손실되는 경우 어떤 순서로 서버를 복구하든 최종 결과에는 영향을 주지 않습니다.

### 시나리오 HASL3.3: 감시 서버 손실에 이어 미러 서버 손실

감시 서버가 실패한 다음 미러 서버가 손실되는 경우 주 서버는 격리됩니다. 안전성이 FULL이고 주 서버가 쿼럼을 구성할 수 없는 경우 그림 10가 같이 더 이상 데이터베이스를 서비스할 수 없습니다.

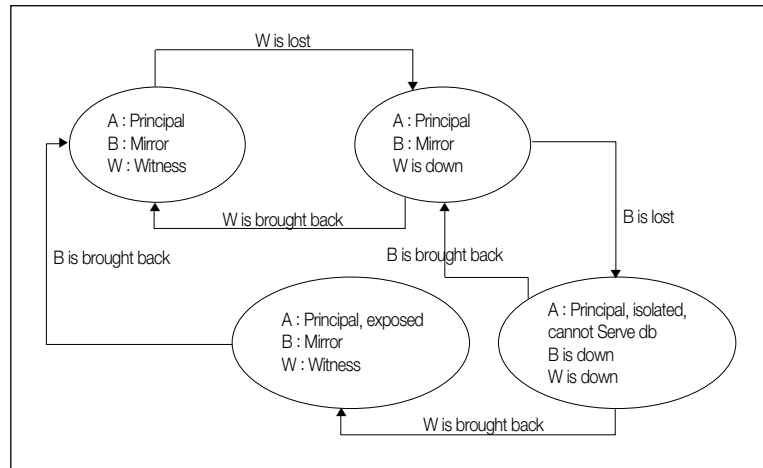


그림 10. 감시 서버가 실패하고 이어서 미러 서버가 실패하면 주 서버는 데이터베이스를 서비스하지 못합니다.

**주:** 서버 W에 이어 서버 B가 손실된 후에 서버 A가 손실되는 경우 어떤 순서로 서버를 복구하든 최종 결과에는 영향을 주지 않습니다.

### 요약: 서버 손실 시의 고가용성 시나리오

이러한 시나리오에서 많은 결론을 이끌어낼 수 있습니다. 고가용성 모드에서는:

주 서버를 먼저 사용할 수 없게 되면 자동 장애 조치가 발생하고 이전 미러 서버가 주 서버 역할을 맡게 되고 사용자는 데이터베이스를 사용할 수 있습니다. 이후의 서버 실패와 복원은 새로운 주 서버의 전체 미러링 구성에 영향을 주지 않습니다. 미러링은 역방향으로 계속됩니다.

미러 서버를 먼저 사용할 수 없게 되면 자동 장애 조치는 발생하지 않습니다. 이후의 서버 비가용성과 복원 순서는 미러링 파트너 역할에 영향을 주지 않습니다.

감시 서버를 먼저 사용할 수 없게 되면 자동 장애 조치는 가능하지 않으며 파트너 서버는 원래 역할을 유지합니다. 이후의 서버 비가용성과 복원은 파트너 역할에 영향을 주지 않습니다.

다음 표는 하나 또는 두 서버가 손실되는 고가용성 시나리오에 대한 결과를 요약합니다. 표에서 가정하는 조건은 안전성이 FULL이고 미러링 세션 서버에 다음과 같은 조건이 있습니다.

A: 주 서버, SYNCHRONIZED  
B: 미러 서버, SYNCHRONIZED  
W: Witness, CONNECTED

표는 장애 조치 시나리오를 회색으로 강조 표시하여 보여줍니다.

**표 10. 파트너 서버 역할과 데이터베이스 상태를 보여주는 단일 및 이중 서버 실패 요약**

| 최초 이벤트               | 쿼럼    | 결과                                            | 중간 상태                                                            | 두 번째 이벤트             | 쿼럼    | 결과            | 최종 상태                                                                        | 첫 번째 서버로 돌아감 | 결과                                                               | 두 번째 서버로 돌아감 | 최종 상태                                                                             |
|----------------------|-------|-----------------------------------------------|------------------------------------------------------------------|----------------------|-------|---------------|------------------------------------------------------------------------------|--------------|------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------|
| A가 손실<br>(HASL1,1)   | B 및 W | 장애 조치:<br>B 서버의<br>데이터베이스<br>실행 중,<br>세션이 표시됨 | A 서버 다운<br>B: 주 서버,<br>DISCONNECTED<br>W: Witness,<br>CONNECTED  | B 서버 다운<br>(HASL1,2) | 없음    | 실행되는<br>서버 없음 | A 서버 다운<br>B 서버 다운<br>W: Witness                                             | A 서버로<br>돌아감 | A: 미러 서버,<br>DISCONNECTED<br>B 서버 다운<br>W: Witness               | B 서버로<br>돌아감 | A: 미러 서버,<br>SYNCHRONIZED<br>B: 주 서버,<br>SYNCHRONIZED                             |
|                      |       |                                               |                                                                  | W 서버 다운<br>(HASL1,3) | 없음    | 실행되는<br>서버 없음 | A 서버 다운<br>B: 주 서버,<br>DISCONNECTED,<br>실행되지 않음<br>W 서버 다운                   | A 서버로<br>돌아감 | A: 미러 서버<br>B: 주 서버<br>W 서버 다운                                   | W 서버로<br>돌아감 | W: Witness,<br>CONNECTED                                                          |
| B is lost<br>HASL2,1 | A 및 W | A 서버의<br>데이터베이스<br>실행 중, 세션<br>이 표시됨          | A: 주 서버,<br>DISCONNECTED<br>B 서버 다운<br>W: Witness,<br>CONNECTED  | A 서버 다운<br>(HASL2,2) | A 및 W | 실행되는<br>서버 없음 | A 서버 다운<br>B 서버 다운<br>W: Witness                                             | B 서버로<br>돌아감 | A 서버 다운<br>B: 미러 서버,<br>DISCONNECTED<br>W: Witness               | A 서버로<br>돌아감 | A 서버로 돌아감<br>A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED                |
|                      |       |                                               |                                                                  | W 서버 다운<br>(HASL2,3) | 없음    | 실행되는<br>서버 없음 | A: 주 서버,<br>DISCONNECTED,<br>실행되지 않음<br>B 서버 다운,<br>W 서버 다운,<br>DISCONNECTED | B 서버로<br>돌아감 | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W 서버 다운 | W 서버로<br>돌아감 | W: Witness,<br>CONNECTED                                                          |
| W is lost<br>HASL3,1 | A 및 B | A 서버의<br>데이터베이스<br>실행 중, 세션<br>이 표시됨          | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W 서버 다운 | A 서버 다운<br>(HASL3,2) | 없음    | 실행되는<br>서버 없음 | A 서버 다운<br>B: 미러 서버,<br>DISCONNECTED<br>W 서버 다운,<br>DISCONNECTED             | W 서버로<br>돌아감 | A 서버 다운<br>B: 미러 서버,<br>DISCONNECTED<br>W: Witness               | A 서버로<br>돌아감 | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W: Witness,<br>CONNECTED |
|                      |       |                                               |                                                                  | B 서버 다운<br>(HASL3,3) | 없음    | 실행되는<br>서버 없음 | A: 주 서버,<br>DISCONNECTED,<br>실행되지 않음<br>B 서버 다운<br>W 서버 다운,<br>DISCONNECTED  | W 서버로<br>돌아감 | A: 주 서버,<br>DISCONNECTED,<br>실행되지 않음<br>B 서버 다운<br>W: Witness    | B 서버로<br>돌아감 |                                                                                   |

## 통신 손실 시의 고가용성 시나리오

고가용성 작동 모드는 세 개의 SQL Server 인스턴스를 필요로 합니다. 서버가 상당한 거리가 있는 둘 또는 세 개의 독립된 사이트에 있을 경우 사이트 간에 통신 문제가 발생할 가능성이 많습니다. 즉, 서버는 사용 가능하지만 통신 회선이 중단될 수 있습니다. 이러한 시나리오는 이전 설정보다 약간 더 복잡하지만 적용되는 원칙은 동일합니다.

통신이 손실된 다음 고가용성 작동 모드 시나리오는 두 집합으로 설명합니다. 첫 번째 세트는 독립된 사이트에 세 개의 SQL Server 인스턴스가 있으므로 세 개의 독립된 통신 회선을 기준으로 합니다. 두 번째 세트는 두 개의 독립된 사이트에 서버가 있고 서버 쌍과 세 번째 서버 사이에 있는 하나의 통신 회선을 기준으로 합니다.

첫 번째 세트로 시작하여 데이터베이스 미러링 세션에 있는 모든 서버 사이에 세 개의 독립된 통신 회선이 있다고 가정합니다. 예를 들어, 주 서버, 미러 서버 및 감시 서버가 세 개의 독립된 사이트에 있을 수 있습니다. 또한 세 개의 서버가 한 사이트에 있지만 개인 네트워크로 연결되어 있을 수도 있습니다.

초기 조건은 서버 A에 주 데이터베이스가 있고 미러링된 데이터베이스가 있는 파트너로 서버 B와 동기화되었다고 가정합니다. 안전성은 FULL로 설정되고 감시 서버(서버 W)는 데이터베이스 미러링 세션의 일부입니다. 그림 11은 초기 구성을 보여줍니다.

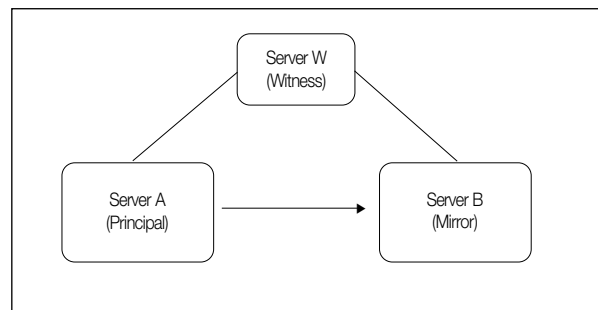


그림 11. 세 개의 독립된 서버에 대한 초기 고가용성 구성은 세 개의 독립된 통신 회선을 갖고 있습니다.

주: 다음 페이지에 있는 다이어그램에 대한 설명은 위의 “서버 손실 시 고가용성 모드”를 참조하십시오.

그림 11을 기준으로 보면 A/B, A/W 및 B/W 등 먼저 연결이 끊어질 수 있는 세 개의 다른 회선이 있습니다. 단일 통신 회선이 다운되면 세 개의 서버 모두 계속 작동합니다. 그림 11에서 볼 수 있듯이 주 서버와 미러 서버 사이의 회선만 영향을 받습니다.

표 11. 단일 통신 회선 연결 끊김의 요약

| 초기 조건                                                                             | 이벤트       | 쿼럼    | 결과                        | 조건                                                                                |
|-----------------------------------------------------------------------------------|-----------|-------|---------------------------|-----------------------------------------------------------------------------------|
| A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W: Witness,<br>CONNECTED | A/B 링크 깨짐 | A 및 W | A 서버의 데이터베이스<br>실행 중, 표시됨 | A: 주 서버,<br>DISCONNECTED<br>B: 미러 서버,<br>DISCONNECTED<br>W: Witness,<br>CONNECTED |
|                                                                                   | A/W       | A 및 B | A 서버의 데이터베이스<br>실행 중      | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W: Witness,<br>CONNECTED |
|                                                                                   | B/W       | A 및 B | A 서버의 데이터베이스<br>실행 중      | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W: Witness,<br>CONNECTED |

주 서버/미러 서버 연결의 끊김만 영향을 받습니다. 주/감시 서버 또는 미러/감시 서버의 다른 끊김은 데이터베이스 미러링 세션의 동작을 변경시키지 않습니다.

요약하면 표 HAC1의 내용은 다음과 같습니다.

- 단일 통신 회선의 끊김 중에서도 주/미러 서버 끊김만 데이터베이스 미러링에 영향을 줍니다. 로그 레코드가 미러 서버로 전송되지 않기 때문에 주 서버는 표시된 상태로 실행됩니다.

이제 두 번째 회선이 끊기는 경우 어떤 일이 발생하는지 생각해 봅시다. 두 회선이 동시에 또는 순서대로 끊길 수 있습니다. 두 회선이 동시에 끊기는 경우 한 회선이 끊긴 다음 다른 회선이 끊기므로 최종 결과는 동일합니다. 그러나 정확한 순서는 사전에 예측할 수 없습니다. 이후의 동작은 동시 끊김이 동일한 순서를 나타냅니다.

따라서 여기서의 목적을 위해 순차적인 회선 끊김만 고려할 것입니다. 표 12는 고가용성 모드에서 통신 회선 끊김의 경우 이 절에서 이름처럼 기본 시나리오를 보여줍니다.

표 12. 두 회선에서 대부분의 통신 회선 끊김은 한 서버가 다운되는 서버 다운 시나리오와 동일해 집니다.

| 시나리오    | 첫 번째 회선 끊김 | 시나리오    | 두 번째 회선 끊김 | 결과       | 나머지 서버에<br>대해 동일한<br>시나리오 | 시나리오 참조 |
|---------|------------|---------|------------|----------|---------------------------|---------|
| HACL1.1 | A/B        | HACL1.2 | A/W        | A 서버 분리됨 | A 서버 다운                   | (없음)    |
|         |            | HACL1.3 | B/W        | B 서버 분리됨 | B 서버 다운                   | HASL2.1 |
| HACL2.1 | A/W        | HACL2.1 | A/B        | A 서버 분리됨 | A 서버 다운                   | HASL1.1 |
|         |            | HACL2.2 | B/W        | W 서버 분리됨 | W 서버 다운                   | HASL3.1 |
| HACL3.1 | B/W        | HACL3.1 | A/W        | W 서버 분리됨 | W 서버 다운                   | HASL3.1 |
|         |            | HACL3.2 | A/B        | B 서버 분리됨 | B 서버 다운                   | HASL2.1 |

표 HACL2가 보여주는 것은 모든 순차적인 두 회선 통신 끊김은 앞 절의 단일 서버 다운 시나리오와 동일하므로 여기에서 분석을 반복하지 않을 것입니다.

중요한 사항은 다음과 같습니다.

- 두 통신이 끊기는 한 시나리오에서만 장애 조치가 발생합니다. 주/감시 서버 회선 끊김에 이어 주/미러 서버 회선이 끊깁니다.

주/미러 서버 회선이 끊기고 이어서 주/감시 서버 회선이 끊기면 주 서버가 격리되었어도 장애 조치는 발생하지 않고 감시 서버가 이를 볼 수 없습니다.

시나리오 HACL1.2를 보다 신중하게 검토해 보겠습니다.

#### 시나리오 HACL1.2: 주/미러 서버 회선이 끊기고 이어서 주/감시 서버 회선이 끊김

주/미러 서버 링크가 끊어지고 이어서 주 서버와 감시 서버 사이의 링크가 끊어지면 주 서버가 격리됩니다. 다른 서버를 볼 수 없으며 쿼리가 손실됩니다. 한편 미러 서버와 감시 서버는 주 서버가 여전히 가동되는지 여부를 알지 못하므로 서버 B는 주 서버의 역할을 맡고 자동 장애 조치가 발생합니다. 그림 12는 이러한 이벤트를 보여줍니다.

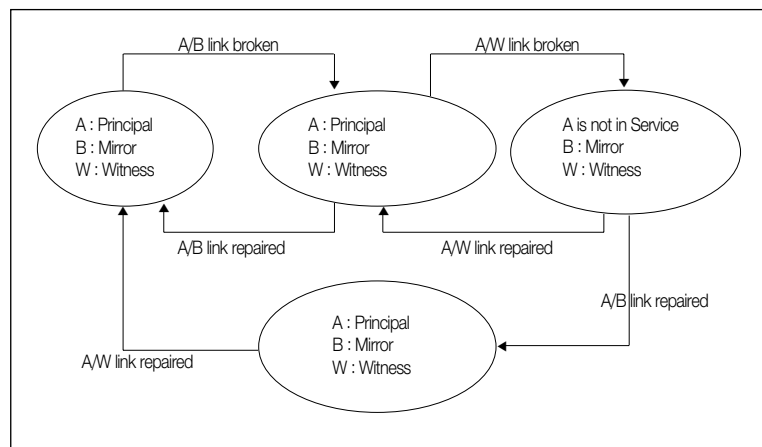


그림 12. 고가용성 모드에서 주/미러 서버 링크가 끊어지고 이어서 주/감시 서버 회선이 끊어지면 장애 조치는 발생하지 않습니다.

주/미러 서버 링크가 끊어지고 이어서 주/감시 서버가 끊어지면 서버 A는 격리되고 데이터베이스를 서비스할 수 없습니다. 서버 A가 서버 B에 없는 작업을 수행할 수 있기 때문에 서버 B와 W는 쿼럼을 구성하지 못합니다.

주/감시(A/W) 서버 회선 끊김이 먼저 복구되면 서버 A는 DISCONNECTED 상태에서 주 서버 역할을 계속합니다. 그러나 주 서버와 미러 서버 사이의 회선은 아직 복구되지 않았기 때문에 미러링은 발생하지 않습니다.

주/미러(A/B) 서버 회선 끊김이 먼저 복구되면 감시 서버가 없어도 서버 A는 서버 B에 대한 미러링을 계속하므로 세션이 표시 됩니다. 그러나 주/감시 서버 회선이 마지막으로 복구될 때까지 자동 장애 조치는 가능하지 않습니다.

#### 요약: 서버 손실 시의 고가용성 시나리오: 세 개의 사이트

다음 표는 세 개의 독립된 서버가 있는 하나 및 두 회선 끊김의 동작을 요약합니다. 표의 초기 조건은 안전성 수준이 FULL이고 서버는 다음과 같은 상태에 있습니다.

A: 주 서버, SYNCHRONIZED

B: 미러 서버, SYNCHRONIZED

W: Witness, CONNECTED

장애 조치 시나리오 경로는 회색으로 강조 표시됩니다.

표 13. 고가용성 모드에서 안전성이 FULL이고 세 개의 독립된 서버의 경우 한 회선 및 두 회선 끊김의 요약

| 최초 이벤트              | 쿼럼    | 결과                     | 중간 상태                                                         | 두번째 이벤트             | 쿼럼    | 결과                              | 최종 상태                                                                          | 복구된 회선 | 첫 번째 복구 상태                                                    |
|---------------------|-------|------------------------|---------------------------------------------------------------|---------------------|-------|---------------------------------|--------------------------------------------------------------------------------|--------|---------------------------------------------------------------|
| A/B 링크 깨짐 (HACL1.1) | A 및 W | A 서버의 데이터베이스 실행 중, 표시됨 | A: 주 서버, DISCONNECTED<br>B: 미러 서버, DISCONNECTED<br>W: Witness | A/W 링크 끊김 (HACL1.2) | B 및 W | 실행되는 서버 없음<br>A 분리됨             | A: 주 서버, DISCONNECTED, db를 서비스할 수 없음<br>B: 미러 서버, DISCONNECTED<br>W: Witness   | A/B    | A: 주 서버, SYNCHRONIZED<br>B: 미러 서버, SYNCHRONIZED<br>W: Witness |
|                     |       |                        |                                                               | B/W 링크 끊김 (HACL1.3) | A 및 W | A 서버의 데이터베이스 실행 중, 표시됨<br>B 격리됨 | A: 주 서버, DISCONNECTED<br>B: 미러 서버, DISCONNECTED<br>W: Witness                  | A/B    |                                                               |
| A/W 링크 끊김 (HACL2.1) | A 및 B | A 서버의 데이터베이스 실행 중      | A: 주 서버, SYNCHRONIZED<br>B: 미러 서버, SYNCHRONIZED<br>W: Witness | A/B 링크 깨짐 (HACL2.2) | B 및 W | B 서버의 데이터베이스 실행 중 표시됨           | A 서버는 실행되지 않음<br>B: 주 서버, DISCONNECTED<br>W: Witness                           | A/W    | A: 미러 서버, DISCONNECTED<br>B: 주 서버, DISCONNECTED<br>W: Witness |
|                     |       |                        |                                                               | B/W 링크 끊김 (HACL2.3) | A 및 B | A 서버의 데이터베이스 실행 중               | A: 주 서버, SYNCHRONIZED<br>B: 미러 서버, SYNCHRONIZED<br>W: 감시 서버, 분리됨, DISCONNECTED | A/W    | A: 주 서버, SYNCHRONIZED<br>B: 미러 서버, SYNCHRONIZED<br>W: Witness |

|                           |       |                              |                                                                     |              |       |                              |                                                                                         |     |                                                                     |
|---------------------------|-------|------------------------------|---------------------------------------------------------------------|--------------|-------|------------------------------|-----------------------------------------------------------------------------------------|-----|---------------------------------------------------------------------|
| B/W 링크<br>끊김<br>(HACL3,1) | A 및 B | A 서버의<br>데이터베이스<br>실행 중, 표시됨 | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W: Witness | A/W 링크<br>깨짐 | A 및 B | A 서버의<br>데이터베이스<br>실행 중      | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W: 감시 서버, 분리됨,<br>DISCONNECTED | B/W | A: 주 서버,<br>SYNCHRONIZED<br>B: 미러 서버,<br>SYNCHRONIZED<br>W: Witness |
|                           |       |                              |                                                                     | A/B 링크<br>깨짐 |       | A 서버의<br>데이터베이스<br>실행 중, 표시됨 | A: 주 서버,<br>DISCONNECTED<br>B: 미러 서버,<br>DISCONNECTED, 분리됨<br>W: Witness                |     | A: 주 서버,<br>DISCONNECTED<br>B: 미러 서버,<br>DISCONNECTED<br>W: Witness |

#### 시나리오 HACL4: 미러 사이트에 감시 서버가 있는 두 사이트

서버 세트 사이에 통신 회선이 하나만 있으면 감시 서버를 배치할 곳을 선택해야 합니다. 먼저 미러 데이터베이스 서버에 감시 서버를 배치한다고 가정합니다. 그림 13과 같이 두 서버 세트 사이에 통신 회선이 하나만 있고 이 회선이 중단될 수 있습니다.

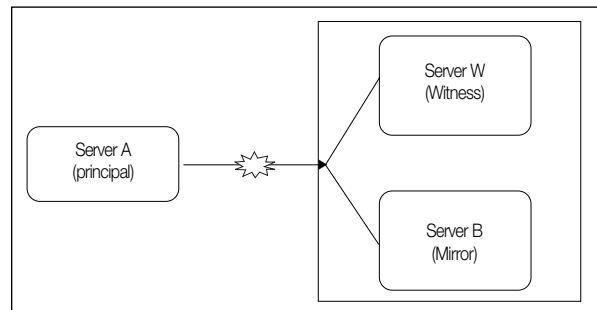


그림 13. 주 서버 및 미러/감시 사이트 사이의 통신 회선이 중단되었습니다.

서버 A는 감시 서버 W 또는 미러 데이터베이스의 서버 B를 볼 수 없으므로 쿼리를 구성할 수 없습니다. 서버 B와 서버 W는 쿼리를 구성할 수 있지만 서버 A에서 주 서버를 볼 수 없습니다. 회선 끊김의 결과는 그림 14와 같습니다.

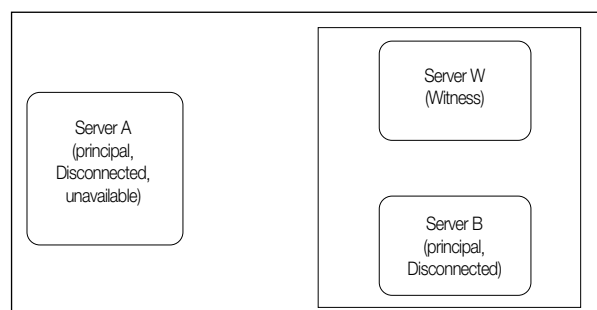


그림 14. 통신이 중단되고 감시 서버가 미러 사이트에 있을 때 장애 조치가 발생합니다.

서버 A는 감시 서버 W 또는 이전 미러 파트너 서버 B를 볼 수 없기 때문에 연결이 끊긴 상태가 되어야 하며 데이터베이스를 사용할 수 없게 됩니다.

서버 B와 서버 W는 쿼럼을 구성할 수 있습니다. 서버 B가 서버 A를 볼 수 없고 서버 W와 서버 B가 주 서버가 되고 데이터베이스가 온라인이 되려고 시도합니다. 서버 W는 서버 A를 볼 수 없기 때문에 서버 B와 동의합니다. 서버 B에는 이제 쿼럼이 있고 이 세션에서 주 서버 역할을 맡고 데이터베이스를 복구합니다.

통신 회선을 복원하는 경우 서버 A는 서버 B가 주 서버인 것으로 확인하고 감시 서버 W가 서버 B를 주 서버로 보는 것을 발견합니다. 서버 A는 미러 서버의 역할로 역할을 변경하고 새로운 주 서버와 동기화를 시도합니다. 완료되면 결과의 구성은 그림 15와 같아집니다.

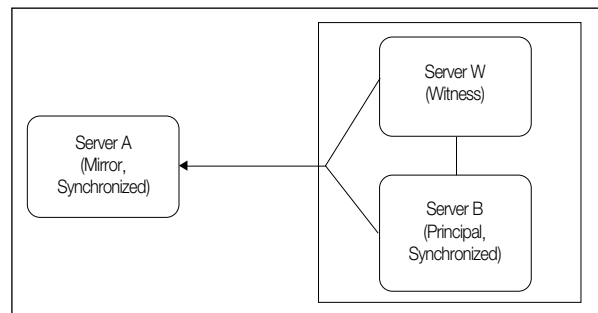


그림 15. 이 시나리오의 복원된 버전은 역방향으로 미러링을 수행합니다.

요약: 감시 서버가 미러 서버와 함께 원격 사이트에 존재하는 경우 사이트 간의 통신 회선이 중단되면 자동 페일오버가 발생합니다.

#### 시나리오 HACLS: 주 사이트에 감시 서버가 있는 두 사이트

이 고가용성 시나리오에서는 그림 16과 같이 감시 서버를 주 데이터베이스 서버와 같은 사이트에 배치하고 두 사이트 사이의 통신이 중단된다고 가정합니다.

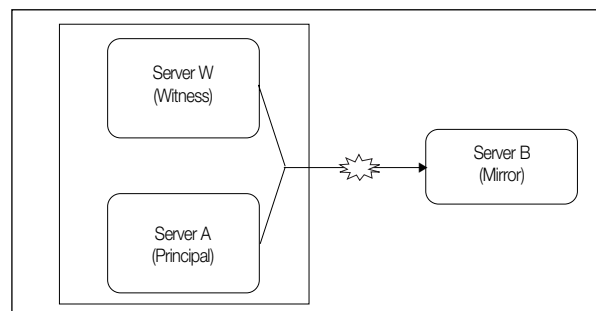


그림 16. 주/감시 서버 및 미러 서버 사이의 통신이 중단됩니다.

이 경우 미러 데이터베이스가 있는 서버 B는 주 서버와 감시 서버로부터 분리됩니다. 서버 A와 서버 W는 쿼럼을 계속 구성하므로 서버 A는 데이터베이스를 주 서버를 유지합니다. 그러나 서버 A는 서버 B를 볼 수 없고 데이터를 전송할 수 없기 때문에 데이터베이스는 연결이 끊긴 상태가 됩니다. 서버 B는 서버 A를 볼 수 없으므로 역시 연결이 끊어진 상태가 됩니다. 결과의 구성은 그림 17과 같습니다.

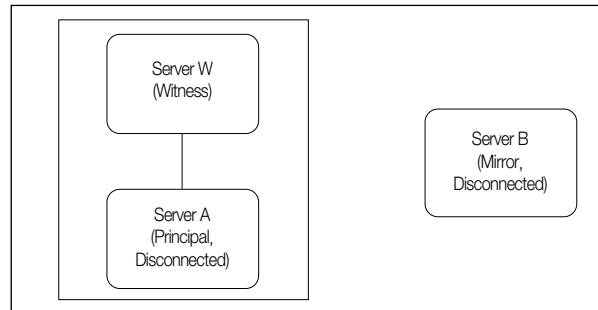


그림 17. 이 시나리오에서 통신이 손실되면 두 파트너의 연결이 끊어지게 됩니다.

서버 A는 트랜잭션을 계속 받지만 트랜잭션 로그를 잘라낼 수는 없습니다. 회선을 신속하게 복원하는 경우 미러링 세션을 다시 동기화하고 원래 작동 상태로 돌아갈 수 있습니다.

요약: 주 서버와 같은 사이트에 감시 서버가 있고 원격 사이트에 미러 서버가 있으면 사이트 사이의 통신 중단은 자동 장애 조치를 일으키지 않습니다.

#### 요약: 통신 링크가 끊긴 상태의 고가용성 시나리오

세 개의 독립된 서버가 있는 고가용성 구성에서 세 개의 독립 통신 회선이 있습니다.

- 주/미러 서버 통신 손실이 발생하면 자동 장애 조치가 발생하지 않습니다.
- 주/감시 서버 통신 손실이 먼저 발생하고 이어서 주/미러 서버 통신이 끊어지면 자동 장애 조치가 발생합니다. 회선을 복원하면 미러링의 역방향을 유지합니다.
- 미러/감시 서버 통신 손실이 발생하면 자동 장애 조치가 발생하지 않습니다.

한 통신 회선만 있는 고가용성 구성에서는 감시 서버는 주 서버 또는 미러 서버에 있습니다.

- 감시 서버가 미러 서버와 함께 원격 사이트에 존재하는 경우 사이트 간의 통신 회선이 중단되면 자동 페일오버가 발생합니다.
- 주 서버와 같은 사이트에 감시 서버가 있고 원격 사이트에 미러 서버가 있으면 사이트 사이의 통신 중단은 자동 장애 조치를 일으키지 않습니다.

## 높은 수준의 보호 시나리오

높은 수준의 보호 작동 모드는 안전성이 FULL에서 작동하지만 감시 서버는 없습니다. 주 데이터베이스가 있는 서버와 미러 데이터베이스가 있는 서버만 관련되기 때문에 통신 회선이 하나만 있습니다. 이는 시나리오 수를 획기적으로 줄입니다.

사례 1. 높은 수준의 작동 모드에서 주 서버와 미러 서버 등 두 SQL Server 인스턴스만 관련됩니다. 감시 서버는 없기 때문에 자동 장애 조치는 가능하지 않습니다. 서버 사이에 통신 회선이 하나만 있고 중단될 수 있어 그림 18과 같은 구성이 가능합니다.

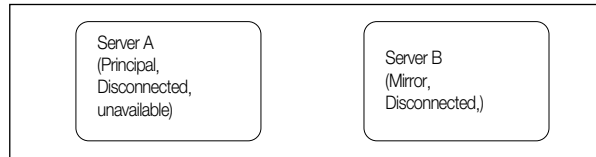


그림 18. 높은 수준의 보호 시나리오에서 통신이 중단되면 두 파트너 모두 연결이 끊어지고 주 서버를 사용할 수 없습니다.

안전성이 FULL이고 주 서버가 미러 서버와 더 이상 쿼럼을 구성할 수 없기 때문에 주 서버는 데이터베이스 연결을 끊어야 하며 사용자 작업에 사용할 수 없습니다.

사례 2. 높은 수준의 보호 시나리오에서 미러 데이터베이스를 사용할 수 없게 되면 주 서버는 그림 19에서 볼 수 있듯이 시나리오에서 동일합니다.

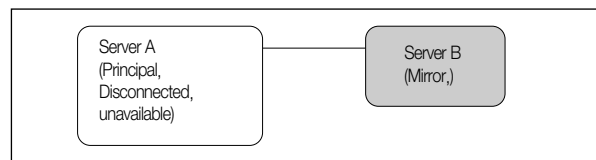


그림 19. 높은 수준의 보호 시나리오에서 미러 서버를 사용할 수 없는 경우 주 데이터베이스가 영향을 받습니다.

사례 3. 높은 수준의 보호 시나리오에서 주 데이터베이스를 사용할 수 없는 경우 미러 데이터베이스는 미러 서버에 유지되어야 하지만 그림 20에서 볼 수 있듯이 연결이 끊긴 상태에 있게 됩니다.

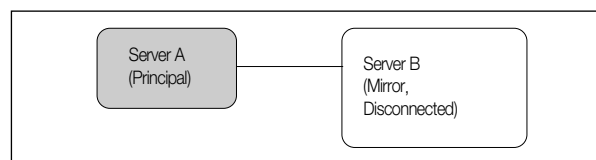


그림 20. 높은 수준의 보호에서 주 서버를 사용할 수 없게 되면 미러 데이터베이스는 연결이 끊긴 상태가 됩니다.

높은 수준의 보호 작동 모드는 안전성이 FULL이기 때문에 중단되면 주 데이터베이스를 사용할 수 없게 되고 미러 데이터베이스는 복구 중 상태에 유지됩니다. 데이터베이스는 온라인 상태가 아닙니다. 결과적으로 고가용성 요구 사항에는 좋은 솔루션이 아닙니다. 짧은 기간 동안 감시 서버를 제거해야 할 때 같이 임시 상태로 훨씬 적합합니다.

## 고가용성 시나리오

고가용성 작동 모드는 안전성이 OFF인 상태에서 작동합니다. 감시 서버에서는 역할이 재생되지 않습니다. 주 데이터베이스가 있는 서버와 미러 데이터베이스가 있는 서버만 관련되기 때문에 통신 회선이 하나만 있습니다. 높은 수준의 보호와 유사함에도 불구하고 고성능은 안전성이 OFF이기 때문에 동작이 다릅니다.

사례 1. 고성능 작동 모드에서 두 SQL Server 인스턴스가 관련이 됩니다. 하나에는 주 데이터베이스가 있고 다른 하나에는 미러 데이터베이스가 포함되어 있습니다. 따라서 서버 사이에 통신 회선이 하나만 있고 중단될 수 있어 그림 21과 같은 구성이 가능합니다.

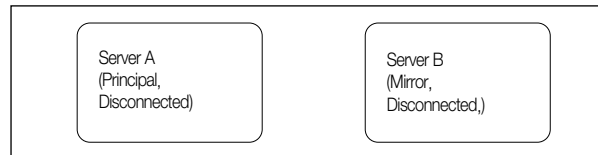


그림 21. 고성능 시나리오에서 통신이 중단되면 두 파트너 모두 연결이 끊어지지만 주 데이터베이스는 사용할 수 있습니다.

트랜잭션 안전성이 OFF이기 때문에 서버 A에는 데이터베이스의 가용성을 유지하기 위해 쿼럼이 요구되지 않습니다. 따라서 연결은 끊어졌지만 주 서버는 사용자 작업을 계속 받아들입니다. 통신을 복원하면 미러 데이터베이스는 추적을 시도하지만 하지 못하거나, 누락된 모든 트랜잭션을 검색할 수 없는 경우 재실행 오류가 발생할 수도 있습니다.

사례 2. 고성능 시나리오에서 미러 데이터베이스를 사용할 수 없게 되면 주 서버의 결과는 그림 22와 같습니다. 안전성이 OFF이므로 주 데이터베이스는 계속 사용할 수 있습니다.

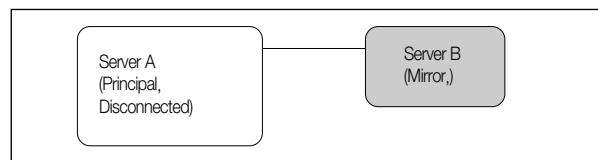


그림 22. 고성능 시나리오에서 미러 서버를 사용할 수 없는 경우 주 데이터베이스는 영향을 받지 않습니다.

사례 3. 높은 수준의 보호 시나리오에서 주 데이터베이스를 사용할 수 없게 되면 미러 데이터베이스는 미러를 유지하지만 그림 23에서 볼 수 있듯이 연결은 끊어집니다.

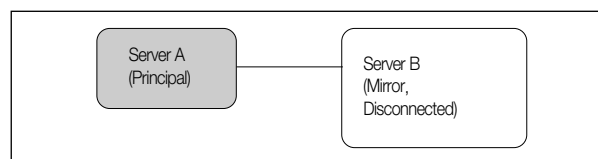


그림 23. 주 서버를 사용할 수 없게 되면 서버 B는 영향을 받지 않지만 연결이 끊긴 상태가 됩니다.

고성능 작동 모드에서는 높은 수준의 보호 모드에서처럼 자동 장애 조치가 가능하지 않습니다. 안전성이 OFF이기 때문에 미러를 사용할 수 없게 되면 주 서버는 데이터베이스를 사용할 수 있도록 유지합니다. 또한 안전성이 OFF이므로 트랜잭션이 미러에 도달하는 것이 보장되지 않습니다. 미러에 강제로 장애 조치하는 경우 일부 트랜잭션은 손실될 수 있습니다.

## 데이터베이스 미러링 구현

데이터베이스 미러링 구현에 대한 기본 정보는 SQL Server 2005 온라인 설명서의 데이터베이스 미러링에 대한 “How To” 항목에서 찾을 수 있습니다. 이 절에서는 몇 가지 최상의 사례와 함께 데이터베이스 미러링 구현의 특정 예를 살펴 보도록 하겠습니다.

### 데이터베이스 미러링 모니터링

SQL Server 2005 Management Studio의 Object Explorer에서 주 또는 미러 데이터베이스를 검토하여 각 데이터베이스 미러링 상태를 감지할 수 있습니다. 주 서버가 미러 서버와 동기화되면 Object Explorer는 주 데이터베이스 이름 옆에 (주 서버, Synchronized) 메시지를 추가하고 미러 서버 이름 옆에 (미러 서버, Synchronized) 메시지를 추가합니다. 주 서버에서 호출하면 Database Properties 대화 상자에 있는 Mirroring 페이지의 Status 상자를 관찰하여 데이터베이스 미러링 세션 상태를 조사할 수도 있습니다. 미러 데이터베이스에서는 Database Properties 대화 상자가 열리지 않습니다.

데이터베이스 미러링 카탈로그 뷰, sys.database\_mirroring 및 sys.database\_mirroring\_witnesses를 쿼리할 수도 있습니다. 카탈로그 뷰를 사용하여 미러링 세션에서 데이터베이스 상태를 조사하는 방법에 대한 자세한 내용은 이 문서 앞부분의 Database Dynamics 절에서 “데이터베이스 미러링 카탈로그 뷰 메타데이터”를 참조하십시오. 카탈로그 뷰는 SQL Server 온라인 설명서에도 자세히 설명되어 있습니다.

### 데이터베이스 미러링 Perfmon 카운터

Perfmon 카운터를 사용하여 데이터베이스 미러링 세션의 파트너 사이에 트래픽을 모니터링할 수 있습니다. SQL Server Database Mirroring 개체에는 주 서버와 감시 서버를 위한 유용한 Perfmon 카운터가 많이 포함되어 있습니다. SQL Server 온라인 설명서의 “Monitoring Database Mirroring”을 참조하십시오.

Database Mirroring 개체의 각 카운터는 데이터베이스 당 설정할 수 있습니다. 서버에서 둘 이상의 데이터베이스를 미러링하는 경우 각 데이터베이스의 활동을 개별적으로 또는 모든 데이터베이스의 전체 미러링 활동을 측정할 수 있습니다.

주 서버의 경우 Log Bytes Sent/sec 카운터는 주 서버가 트랜잭션 로그 데이터를 미러 서버로 전송하는 속도를 알려주는 반면, Log Send Queue는 어느 한 시점에 트랜잭션 로그 버퍼에서 어느 정도의 바이트를 미러 서버로 전송되는지 보여줍니다. 트랜잭션 로그 데이터를 주 서버에서 미러 서버로 전송하면 주 서버의 전송 대기열이 고갈되지만 새 로그 레코드가 로그 버퍼에 들어오면 크기가 커집니다. 주 서버의 Transaction Delay 카운터는 미러 데이터베이스가 수신하는 로그 레코드의 승인을 기다리므로 주 서버에 발생하는 지연을 보여줍니다. Pages Sent/sec는 동기화를 지원하기 위해 데이터베이스를 미러 서버로 전송하는 주 서버와 관련된 것입니다.

미러 서버에서 Log Bytes Received/sec는 미러가 주 서버와 얼마나 상태를 잘 유지하는지 보여줍니다. 위의 Log Bytes Sent/sec 카운터를 참조하십시오. Redo Queue 카운터는 진행 중인 재실행 단계 동안 미러 서버가 주 서버에서 트랜잭션을 재실행하는 데 사용하는 재실행 대기열 크기를 보여줍니다. Redo Bytes/sec는 미러 서버가 재실행 대기열에서 이러한 트랜잭션을 재실행하는 속도를 보여줍니다.



각 파트너에 대해 Sends/sec and Receives/sec 카운터는 서버가 통신하는 속도에 대해 알 수 있도록 개별 전송 및 수신 작업 수를 보여줍니다. Bytes Sent/sec 및 Bytes Received/sec 카운터는 이러한 전송과 수신에 대해 각 파트너 서버에서 전송하고 수신한 총 바이트 수를 보여줍니다.

### 재실행 및 추적 시간 예측

Redo Queue 및 Redo Bytes/sec의 값을 사용하여 미리 데이터베이스가 재실행을 완료하고 사용할 수 있게 되며 장애 조치가 발생하는 데 걸리는 시간을 예측할 수 있습니다. 예측은 다음과 같은 간단한 수식으로 구성됩니다.

$$\text{재실행 예측 시간(초)} = \frac{(\text{Redo Queue})}{(\text{Redo Bytes/sec})}$$

마찬가지로 Log Send Queue 및 Log Bytes Received/sec 카운터를 사용하여 주 서버의 활동이 앞서 있고 이후로 다운된 경우 미러가 주 서버를 추적하는 데 걸리는 시간을 예측할 수 있습니다. 예측은 다음 수식을 통해 가능합니다.

$$\text{추적 예측 시간(초)} = \frac{(\text{Log Send Queue})}{(\text{Log Bytes Received /sec})}$$

### Profiler 이벤트

SQL Server 2005 Profiler에는 데이터베이스 미러링을 위한 하나의 이벤트 클래스가 포함되어 있습니다. Database:Database Mirroring State Change 이벤트는 서버가 상태 변경을 모니터링하는지 여부를 기록합니다. SQL Server 온라인 설명서의 “Database Mirroring State Change Event Class” 항목을 참조하십시오. 이 이벤트 클래스를 사용할 때 Database Name과 State 열을 포함하는 것이 유용합니다. 이 이벤트를 사용하여 데이터베이스 미러링 세션에서 상태 변경을 경고할 수 있습니다.

## 데이터베이스 미러링 문제 해결

오류 가능성이 가장 많은 두 영역은 설치와 실행 시간 동안입니다.

### 설치 문제 해결

데이터베이스 미러링을 설치했지만 시작하지 않은 경우 설치 동안 수행한 단계를 다시 추적하십시오.

미러 서버가 주 서버를 닫을 충분한 시간이 있는지 확인하십시오. 모니터링을 시작할 때 다음 메시지가 나타나는 경우

데이터베이스 “AdventureWorks”의 원격 복사본이 데이터베이스 로그의 로컬 복사본에 포함된 시점으로 롤 포워드되지 않았습니다. (Microsoft SQL Server, 오류: 1412)

미러가 추적되지 않은 것을 알고 있습니다. 주 서버에서 로그 레코드를 받을 수 있는 지점으로 미러를 추적하기 위해 주 서버에서 미러 서버로 트랜잭션 로그 백업을 적용할 필요가 없습니다(NORECOVERY 사용).

각 서버의 SQL Server Windows 서비스 계정이 다른 서버에서 신뢰되는지 확인하십시오. 서버가 신뢰되지 않는 도메인에 있는 경우 인증서가 올바른지 확인하십시오.

sys.database\_mirroring\_endpoints 카탈로그 뷰를 쿼리하여 중단점이 정의되지 않았지만 시작되었는지 확인하십시오.

```
SELECT * FROM sys.database_mirroring_endpoints;
```

또한 정규화된 컴퓨터 이름이 올바르며 포트 번호가 올바른지 확인하십시오. 단일 실제 서버에 있는 인스턴스에서 미러링하는 경우 포트 번호는 고유해야 합니다. 각 SQL Server 서비스 로그인은 중단점에 대한 CONNECT 권한이 있어야 합니다.

마지막으로 의도한 역할에 맞게 서버에 대해 각 중단점 역할을 적절히 정의했는지 확인하십시오.

ALTER DATABASE 명령에서 올바른 파트너 이름을 식별했는지 확인하십시오. 주 서버와 미러 서버에서 sys.database\_mirroring 카탈로그 뷰에서 파트너 이름을 조사할 수 있습니다 (고가용성 모드에 있는 경우 sys.database\_mirroring\_witnesses witness).

### 실행 시간 오류 문제 해결

데이터베이스 미러링을 올바르게 설치했고 실행 중에 오류가 발생하는 경우 세션의 현재 상태를 확인하십시오. 오류로 인해 SUSPENDED 상태가 된 경우 미러에서 재실행 오류가 발생했을 수 있습니다. 재실행(데이터 드라이브의 여유 공간) 및 로그 저장(로그 드라이브의 여유 공간)을 위한 충분한 디스크 공간이 미러 서버에 있는지 확인하십시오. 세션을 다시 시작할 다시 시작할 준비가 되면 ALTER DATABASE를 사용하여 세션을 RESUME하십시오.

주 데이터베이스에 연결할 수 없는 경우 안전성이 FULL이고 주 서버가 쿼럼을 구성할 수 없기 때문일 수 있습니다. 예를 들어, 이는 시스템이 높은 수준의 보호 모드에 있고(안전성이 FULL이지만 감시 서버는 없음) 미러는 미러 서버가 이전의 주 서버에서 연결이 끊어진 경우 발생할 수 있습니다. 미러에서 다음 명령을 사용하여 미러 서버를 강제로 복구합니다.

```
ALTER DATABASE [AdventureWorks] SET PARTNER FORCE_SERVICE_ALLOW_DATA_LOSS
```

문제는 미러 데이터베이스를 복구한 후에 주 서버가 되고 쿼럼을 구성할 수 없으므로 데이터베이스를 서비스할 수 없다는 것입니다. 이 경우 안전성을 OFF로 설정하여 데이터베이스를 서비스할 수 있습니다.

## 안정성 대 성능

데이터베이스 미러링 성능은 활동 유형과 트랜잭션 안정성 설정의 함수입니다.

주 서버의 성능은 미러 서버에 대한 로그 레코드의 전송에 영향을 받습니다. 데이터베이스 미러링이 주 서버에 주는 오버헤드는 활동 유형의 함수입니다. 데이터베이스 미러링은 데이터베이스 서버의 정상 트랜잭션 활동이 로그 레코드를 미러 서버에 전송하는 오버헤드를 마스크하기 때문에 많은 사용자의 많은 긴 트랜잭션에서 가장 잘 수행됩니다. 단일 사용자가 많은 수의 작은 트랜잭션을 순서대로 수행하면 각 트랜잭션에 대한 데이터베이스 미러링의 오버헤드는 더 두드러집니다.

주 서버의 성능은 안전성 설정에 의해서도 영향을 받습니다. 안전성이 FULL이면 주 서버는 트랜잭션 커밋 메시지를 다시 클라이언트에 실행하기 전에 로그 레코드를 수신한 미러에 의해 승인을 기다려야 합니다. 많은 사용자와 긴 트랜잭션에서 이 오버헤드는 중요하지 않습니다. 단일 스레드와 많은 작은 트랜잭션이 있는 시스템은 안전성이 OFF인 상태에서 더 잘 작동할 수 있습니다.

미러 서버가 데이터 수정 트랜잭션을 계속 재생하기 때문에 미러 서버의 데이터 캐시는 '새로운' 데이터입니다. 즉, 데이터 캐시는 주 서버에서 이루어진 같은 종류의 변경을 기반으로 데이터와 인덱스 페이지로 채워집니다. 미러 캐시를 주 서버 캐시와 비슷하게 만들려면 데이터베이스 미러링이 SELECT 힌트를 미러에 전달하여 데이터 쿼리에 사용되는 캐시가 미러 서버에



서 재생성되도록 합니다. 이는 미리 서버를 주 서버와 비슷하게 만들고 장애 조치의 경우 나머지 재실행 시간을 줄이는 데 도움을 줍니다. 명확하게 데이터베이스 스냅샷에 대한 쿼리를 포함하여 미리 서버의 추가 활동은 캐시의 상태에 영향을 미치고 장애 조치의 경우 재실행 단계를 완료하는 기간을 증가시킬 수 있습니다.

## 데이터베이스 미러링 테스트

데이터베이스 미러링을 테스트하는 자체 시스템을 설치할 때 여러 가지 옵션을 사용할 수 있습니다. 모든 미러링은 데이터베이스 미러링 세션의 서버가 개별 SQL Server 인스턴스여야 합니다. 따라서 SQL Server 2005의 관련 엔진의 여러 인스턴스를 설치하는 경우 단일 실제 서버에서 데이터베이스 미러링에 대해 배우고 테스트할 수 있습니다. 단일 가상 서버에서 여러 인스턴스를 테스트할 수도 있지만 실제 서버에서 수행하는 경우 테스트는 더욱 안전해집니다.

로드나 스트레스를 목적으로 데이터베이스 미러링을 테스트할 때는 별개의 실제 서버가 필요합니다. 단일 서버에 인스턴스가 둘 또는 세 개 있으면 실제 서버의 리소스를 비현실적으로 공유할 수 있습니다. 동일하게 중요한 것은 서버 사이의 적절한 연결입니다. 주 서버와 미리 서버 사이의 네트워크 연결이 좋을 수록 로그 레코드와 메시지의 전송 속도가 더 나아집니다.

가장 현실적인 테스트는 실제 대상 서버 또는 최종 시스템과 같은 실제 속성을 갖고 있는 테스트 베드에서 수행됩니다. 단일 서버에서 여러 인스턴스를 테스트할 때 인스턴스 중 하나를 중지하거나 시스템을 종료하여 데이터베이스 미러링에서 서버 손실의 효과를 시뮬레이션만할 수 있습니다. 여러 실제 서버가 있으면 네트워크 케이블의 연결을 끊어 연결 손실을 테스트할 수도 있습니다.

다음 사례는 테스트 조건을 만드는 데 도움을 줄 수 있습니다.

- 서버 오류를 테스트하려면 SQL 구성 관리자를 통하거나 SHUTDOWN WITH NOWAIT를 사용하여 SQL Server 인스턴스를 종료하십시오.
- 통신 오류를 테스트하려면 서버에서 네트워크 케이블을 제거합니다.
- 데이터베이스 오류를 테스트하려면 SQL Server 서버를 테스트하고 기본.mdf 파일 이름을 바꾸려면 SQL Server를 다시 시작합니다.
- 미리에서 재실행 오류를 일으키려면 미리 서버에 없는 드라이버 볼륨의 주 데이터베이스에 파일을 추가합니다.
- 미리 서버에서 재실행 오류를 일으키는 또 다른 방법은 미리 서버 데이터 파일을 공간이 부족한 디스크에서 실행시키는 것입니다.
- 주 서버에서 데이터베이스를 강제로 종료하려면 주 서버 데이터 파일을 공간이 부족한 디스크에서 강제로 시작합니다.
- 로그 버퍼 저장에 주 서버 또는 미리 버퍼에서 강제 실행되도록 하려면 로그 파일을 공간이 부족한 디스크에서 강제로 실행합니다.

## 미리 서버의 장애 조치 준비

데이터베이스 미러링은 엄격히 데이터베이스 간 관계입니다. 데이터베이스 데이터만 주 서버에서 미리 서버로 로그 레코드를 사용하여 전송됩니다. 로그 전달과 복제와 마찬가지로 장애 조치의 경우 완벽하게 역할을 맡도록 미리 데이터베이스를 사용하여 대기 서버를 준비해야 합니다. 미리 서버를 준비할 때 고려할 수준은 여러 가지가 있습니다.

실제 서버 수준에서 대기 서버는 가능한 유사한 실제 CPU와 메모리 구성을 갖고 있어야 하며, 그렇지 않으면 대기 서버는 장애 조치 후에 수행됩니다. 데이터베이스 응용 프로그램을 지원하는 지원 응용 프로그램, 모니터 또는 기타 실행 파일도 있을 수 있으며 미리 서버에서 구성해야 합니다.

SQL Server 수준에서 대기 서버는 같은 SQL Server 구성(예: AWE, max degree of parallelism)을 갖고 있어야 합니다. 그러나 가장 기본적인 것은 로그인과 해당 권한이 될 것입니다. 주 서버의 모든 활성 SQL Server 로그인은 미리 서버에도 있어야 합니다. 그렇지 않으면 응용 프로그램은 장애 조치의 경우 새로운 주 서버로 사용할 수 없게 됩니다. SQL Server Integration Services에는 한 서버에서 다른 서버로 로그인과 암호를 복사하는 데 사용할 수 있는 Transfer Logins 작업이 있지만 이러한 로그인에 대한 데이터베이스 권한을 설정해야 할 수 있습니다. 다른 도메인에 있는 서버로 로그인을 전송하는 경우 SID가 일치하지 않을 수 있으며 일치시켜야 합니다.

대기 서버로 마이그레이션해야 할 주 SQL Server에 여러 가지 지원 개체가 있을 수 있습니다. 몇 가지를 열거해 보면 SQL Agent 작업과 경고, SQL Server Integration Services 패키지, 지원 데이터베이스, 링크된 서버 정의, 백업 서비스, 유지 관리 계획, SQL Mail 또는 Database Mail 설정, Distributed Transaction Coordinator 설정 등이 있습니다.

SQL Agent 작업을 대기 서버로 전송하면 대부분 사용하지 않아야 합니다. 장애 조치의 경우 이러한 작업을 사용해야 합니다.

장애 조치 후에 응용 프로그램이 SQL Server 인증을 사용하는 경우 새로운 주 데이터베이스의 사용자가 새로운 주 SQL Server에서 로그인을 확인해야 합니다. 이를 위한 가장 좋은 도구는 저장 프로시저 sp\_change\_users\_login입니다.

## 다중 데이터베이스 문제

많은 응용 프로그램이 단일 서버에서 여러 데이터베이스를 사용합니다. 한 응용 프로그램이 여러 데이터베이스를 참조하거나 모든 응용 프로그램이 여러 데이터베이스를 사용할 수도 있습니다. 그러나 데이터베이스 미러링은 한 번에 한 데이터베이스만 사용합니다. 데이터베이스 아키텍처에 미러링을 설계할 때는 이것을 고려해야 합니다.

고가용성 모드가 필요할 경우 가장 좋은 방법은 한 응용 프로그램을 한 데이터베이스와 일치시키는 것입니다. 자동 장애 조치가 발생하는 경우 응용 프로그램은 더 이상 주 서버에 데이터베이스를 요구하지 않습니다. 단일 서버에 여러 데이터베이스가 있고 고가용성 모드에서 작동하는 경우 어떤 일이 발생할 수 있는지 고려하십시오. 실제 서버 중단, SQL Server 인스턴스 오류 또는 통신 오류가 발생한 경우 모든 데이터베이스는 대기 서버로 자동으로 장애 조치되며 미리 서버는 새로운 주 데이터베이스가 됩니다. 감시 서버를 볼 수 있는 경우 응용 프로그램은 새로운 주 데이터베이스에 연결할 수 있습니다. 그러나 데이터베이스 중 하나에서 디스크 오류로 손상된 페이지가 발생하여 한 데이터베이스만 장애 조치된 경우 어떤 일이 발생할까요? 이 경우 응용 프로그램이 적절한 모든 데이터베이스에 연결하는 것이 불가능할 수 있습니다.

따라서 여러 데이터베이스에 의존하는 응용 프로그램은 데이터베이스 미러링의 고가용성 모드에 적합하지 않을 것입니다. 자동 장애 조치가 없지만 다른 데이터베이스 서버가 동기화를 유지하는 고성능 방법이 없다는 것을 인식하여 안전성 OFF를 사용할 수 있습니다.

## 데이터베이스 미러링 및 고가용성 기술

SQL Server 2005는 이제 최소한 4개의 고가용성 기술이 있고 일부 중복되기는 하지만 각 기술마다 상대적인 장점과 단점이 있습니다. 이러한 기술은 다음과 같습니다.

- **데이터베이스 미러링** - 설명을 위해 안전성이 FULL이고 감시 서버가 있는 고가용성 작동 모드를 고려할 것입니다.
- **장애 조치 클러스터링** - 가장 일반적인 구성은 SQL Server 인스턴스가 하나인 두 노드의 Windows 장애 조치 클러스터입니다.
- **로그 전달** - 별도의 모니터링이 있는 SQL Server 기본 제공 로그 전달을 가정합니다.
- **트랜잭션 복제** - 게시 서버가 실패하는 경우 대기 서버의 역할을 하는 별도의 배포 서버와 하나의 가입자가 있는 구성을 고려하십시오.

이 절에서는 이러한 네 가지 기술의 기본 기능을 비교하고 데이터베이스 미러링이 더 나은 솔루션을 보완하거나 입증할 수 있는 영역을 자세히 살펴 보겠습니다.

다음 표는 네 가지 기술의 여러 가지 가용성 기능을 보여줍니다.

표 14. SQL Server 2005 고가용성 기술 비교

| 카테고리        | 가용성 특징            | 데이터베이스 미러링(HA 모드)         | 장애 조치 클러스터링     | 로그 전달            | 트랜잭션 복제장애 조치   |
|-------------|-------------------|---------------------------|-----------------|------------------|----------------|
| 장애 조치 특성    | 대기 유형             | Hot                       | Hot             | Warm             | Hot            |
|             | 자동 역할 변경          | 예.                        | 예.              | 사용자 정의 코딩 필요     | 사용자 정의 코딩 필요   |
|             | 장애 조치가 커밋된 작업을 보존 | 예.                        | 예.              | 아니오.             | 아니오.           |
|             | 장애 조치 유형          | 자동 및 수동                   | 자동 및 수동         | 변수               | 변수             |
|             | 장애 조치 동안          | 10초 미만                    | 30초 +           |                  |                |
|             | 데이터베이스            |                           | 데이터베이스 복구       |                  |                |
|             | 다운 시간             |                           |                 |                  |                |
| Physical 구성 | 중복 저장 위치          | 예.                        | 아니오(공유 디스크)     | 예.               | 예.             |
|             | 하드웨어 요구 사항        | 표준 서버                     | 클러스터 인증 서버 및 저장 | 표준 서버            | 표준 서버          |
|             | 실제 거리 제한          | 없음                        | 100 마일          | 없음               | 없음             |
|             | 추가 서버 역할          | Witness                   | 없음              | 모니터              | Distributor    |
| 관리          | 복잡도               | 낮음                        | 높음              | 낮음               | 중간             |
|             | 대기 서버 액세스 가능      | 데이터베이스 스냅샷을 통해; 가능한 성능 영향 | 아니오.            | R/O, 복원과 호환되지 않음 | 읽기 전용 작업의 경우 예 |

|           |            |                                  |              |              |              |
|-----------|------------|----------------------------------|--------------|--------------|--------------|
|           | 여러 보조      | 아니오.                             | 아니오.         | 예.           | 예.           |
|           | 보조에서 로드 지연 | 아니오.                             | 아니오.         | 예.           | 아니오.         |
|           | 가용성 범위     | Database                         | 서버 인스턴스      | Database     | Database     |
| 클라이언트 액세스 | 클라이언트 리디렉션 | ADO.NET 및 SQL Native Client에서 지원 | 필요 없음, 가상 IP | 사용자 정의 코딩 필요 | 사용자 정의 코딩 필요 |

위의 표는 네 가지 가용성 기술의 여러 가지 특성을 요약한 것입니다. 다음 절에서는 몇 가지 자세한 비교를 제공합니다.

## 데이터베이스 미러링 및 클러스터링

데이터베이스 미러링과 장애 조치 클러스터링 사이의 가장 중요한 차이는 각각이 제공하는 중복의 수준입니다. 데이터베이스 미러링은 데이터베이스 수준의 보호를 제공하는 반면, 클러스터링은 서버 인스턴스 수준의 보호를 제공합니다. 다른 중요한 차이점은 데이터베이스 미러링에서 주 서버와 미러 서버가 서로 다른 이름을 가진 별개의 SQL Server 인스턴스인 반면, 클러스터의 SQL Server 인스턴스는 클러스터의 어떤 노드가 인스턴스를 호스팅하는지에 관계 없이 동일하게 유지되는 하나의 가상 서버 이름과 IP 주소를 얻는다는 것입니다.

서버 수준에서 데이터베이스를 보호해야 하는 경우(예를 들어, 응용 프로그램이 같은 데이터베이스 서버에서 동시에 많은 데이터베이스에 액세스해야 하는 경우), 장애 조치 클러스터링이 더 적절한 선택일 수 있습니다. 그러나 한 번에 한 데이터베이스에 대한 가용성을 제공하도록 연결된 경우 데이터베이스 미러링은 여러 가지 장점을 갖고 있습니다.

클러스터링과 달리 데이터베이스 미러링은 전용 하드웨어가 필요하지 않으며 공유 저장 공간이 있는 잠재적인 오류 지점이 없습니다. 데이터베이스 미러링은 다른 고가용성 기술보다 훨씬 빨리 대기 데이터베이스를 서비스로 가져오며 클라이언트쪽 장애 조치의 경우 ADO.NET 및 SQL Native Access Client의 새로운 기능에서 잘 작동합니다.

클러스터 내에서 데이터베이스 미러링을 사용할 수 없지만 클러스터 인스턴스 데이터베이스의 hot 대기 서버를 만드는 방법으로 데이터베이스 미러링 사용을 고려할 수 있습니다. 이렇게 하면 클러스터 장애 조치가 데이터베이스 미러링의 시간 초과 값보다 길기 때문에 고가용성 모드 미러링 세션은 주 서버에 오류가 발생할 때 클러스터 장애 조치에 반응합니다. 클러스터 노드가 미러링 상태가 됩니다.

## 데이터베이스 미러링 및 트랜잭션 복제

데이터베이스 미러링 및 트랜잭션 복제는 모두 원본 서버의 트랜잭션 로그 읽기를 기준으로 하지만 기술은 상당히 다릅니다. 트랜잭션 복제에 대한 자세한 내용은 SQL Server 온라인 설명서의 관련 항목을 참조하십시오. 트랜잭션 복제는 게시자 데이터베이스의 사용자 트랜잭션을 몇 초 만에 가입자에 전달할 수 있기 때문에 고가용성에 주로 사용됩니다. 데이터베이스 미러링은 복제만큼 또는 복제보다 더 빠르다는 장점이 있지만 사용자 테이블과 관련된 트랜잭션만이 아니라 모든 데이터베이스의 트랜잭션을 전달합니다.

트랜잭션 복제는 보고를 위해 여러 게시자에게 데이터를 확장하는 적절한 기술입니다. 트랜잭션 복제 가입자 데이터베이스는 일반적으로 읽기 전용으로 간주되므로 거의 실시간 데이터 액세스가 필요할 때 이상적입니다.

데이터베이스 미러링은 트랜잭션 복제와 호환되며 게시자 데이터베이스의 hot 대기를 유지하는 방법으로 가장 유용합니다. 로그 전달 같은 복제 게시자를 보호하는 다른 방법은 게시자 자체 가입자에 앞서 게시자를 위해 대기 서버를 유지할 수 없습니다. 즉, 트랜잭션 복제는 트랜잭션 로그 백업 스키마보다 훨씬 빠르게 가입자에게 트랜잭션을 전달할 수 있습니다. 데이터베이스 모니터링이 아주 빠르므로 게시자 데이터베이스의 hot 대기를 유지하는 데 훨씬 적합합니다. 그러나 게시자가 실패하는 경우 복구된 대기 데이터베이스를 게시자로 수동으로 다시 설정하고 배포 서버에 다시 연결해야 합니다. 따라서 게시자 서버를 대기 서버로 유지하기 위해 로그 전달을 사용하는 경우 현재 수행해야 합니다.

## 데이터베이스 미러링 및 로그 전달

데이터베이스 미러링 및 로그 전달은 SQL Server 데이터베이스의 복원과 복구 기능에 의존합니다. 데이터베이스 미러링 미러 데이터베이스는 계속 복구 중 상태에 있으며, 주 서버에서 트랜잭션을 계속해서 재생합니다. 로그 전달 보조 서버는 트랜잭션 로그 백업에서 주기적으로 적용된 트랜잭션을 재생합니다. 대량 로그 데이터가 트랜잭션 로그 백업에 추가하기 때문에 로그 전달은 대량 로그 복구 모델에서 작동할 수 있습니다. 반면에 데이터베이스 미러링은 주 서버에서 미러 서버로 로그 레코드를 직접 전송하며 대량 로그 데이터를 전달할 수 없습니다.

많은 경우 데이터베이스 미러링은 고가용성 및 자동 장애 조치의 로그 전달과 같은 종류의 데이터 중복을 제공할 수 있습니다. 그러나 응용 프로그램이 한 서버의 여러 데이터베이스에 의존하는 경우 로그 전달은 동일하게 유효한 방법일 수 있습니다. 앞 절의 “다중 데이터베이스 고려 사항”을 참조하십시오.

또한 로그 전달이 가용성을 보충할 수 있는 데이터베이스 미러링 시나리오가 있습니다. 예를 들어, 고가용성 데이터베이스 미러링 구성을 가질 수 있으며 재해 복구 목적을 위해 주 서버를 원격 사이트에 로그 전달할 수 있습니다. 그림 24는 이러한 구성이 발생할 수 있는 방법을 보여줍니다.

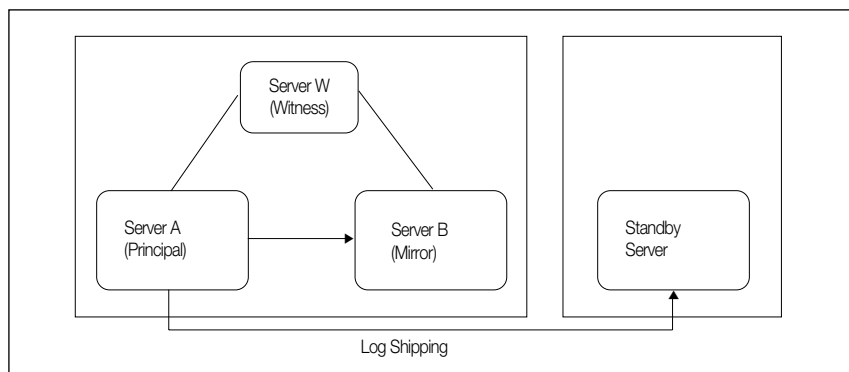


그림 24. 원격 위치로 주 데이터베이스를 로그 전달할 수 있습니다.

여기의 장점은 전체 사이트의 손실이 발생하는 경우 보조 사이트에서 데이터를 사용할 수 있습니다. 그러나 데이터베이스 미러링 장애 조치의 경우 서버 B에서 원격 대기 서버로의 로그 전달은 일반적으로 다시 초기화해야 합니다.

데이터베이스 미러링을 보완하기 위해 로그 전달을 사용하는 또 다른 시나리오는 데이터베이스 미러링 세션이 재해 복구에 사용되는 주 서버에 대한 로컬 대기 서버로 사용되는 것입니다. 이 경우 미러링 세션은 원격 사이트의 미러 서버가 원격 대기 서버인 고성능 모드에 있습니다.

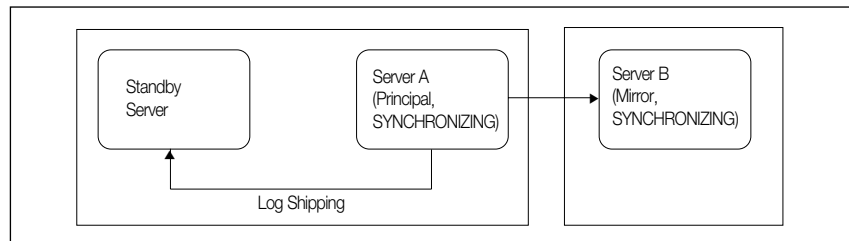


그림 25. 모든 트랜잭션을 유지하는 방법으로 주 데이터베이스를 로그 전달할 수 있습니다.

고성능 모드에서 주 서버가 실패하고 미러 서버가 강제 서비스 복구를 사용하여 복구된 경우 데이터 손실 가능성이 있습니다. 이전 주 서버를 로그 전달하고 이전 주 서버의 트랜잭션 로그 파일이 손상되지 않은 경우 주 서버의 'tail of the log' 백업을 만들어 트랜잭션 로그에서 로그 레코드의 마지막 세트를 얻을 수 있습니다. 대기 로그 전달 데이터베이스에 다른 모든 트랜잭션 로그 백업이 적용된 경우 대기 서버에 로그 백업의 끝을 적용할 수 있으며 이전 주 서버의 데이터가 손실되지 않습니다. 그런 다음 로그 전달 대기 서버의 데이터를 원격 데이터베이스와 비교하고 누락된 데이터를 원격 서버로 복사할 수 있습니다.

모든 경우에 로그 전달을 데이터베이스 미러링 비교는 주 데이터베이스의 데이터베이스와 트랜잭션 로그 백업을 유지하는 데 중요하다는 것은 명백합니다. 이러한 로그 백업을 로그 전달 서버에 적용하면 데이터베이스 미러링 구성을 보완할 수 있습니다.

## 결론

데이터베이스 미러링은 데이터베이스 중복을 위한 고가용성과 고성능 솔루션을 전달할 수 있는 새로운 SQL Server 2005 기술입니다. 데이터베이스 미러링에서 주 서버의 트랜잭션 로그 버퍼가 디스크에 기록될 때마다 트랜잭션 로그 레코드는 주 서버에서 미러 데이터베이스로 직접 전송됩니다. 이 기술은 주 서버와 거의 최신 상태로 미러 데이터베이스를 유지할 수 있으며 커밋된 데이터는 손실되지 않습니다. 고가용성 작동 모드에서는 주 서버가 실패하는 경우 미러 서버는 자동으로 새로운 주 서버가 되고 데이터베이스를 복구합니다. 새로운 ADO.NET 또는 SQL Native Access Client 드라이버를 사용하여 응용 프로그램은 클라이언트 서버에서도 자동 장애 조치를 수행할 수 있습니다. 데이터베이스 미러링은 SQL Server 2005에서 지원하는 고가용성 기술에서 중요한 새 옵션이 되고 있습니다.