**Proje 4: Veritabanı Yük Dengeleme ve Dağıtık Veritabanı Yapıları**

**Kullanılan veri tabanı:** Powerlifting Database

**Link:** <https://www.kaggle.com/datasets/open-powerlifting/powerlifting-database?resource=download>

**Video Linki:** [**https://drive.google.com/file/d/1G4Lj\_5a1t-lyGJpAR0S05I0RsrhekFbJ/view?usp=sharing**](https://drive.google.com/file/d/1G4Lj_5a1t-lyGJpAR0S05I0RsrhekFbJ/view?usp=sharing)

Bu projede istenen adımlar 3 temel başlık altında değerlendirilmiş.

1. Veritabanı Replikasyonu
2. Yük Dengeleme
3. Failover Senaryoları

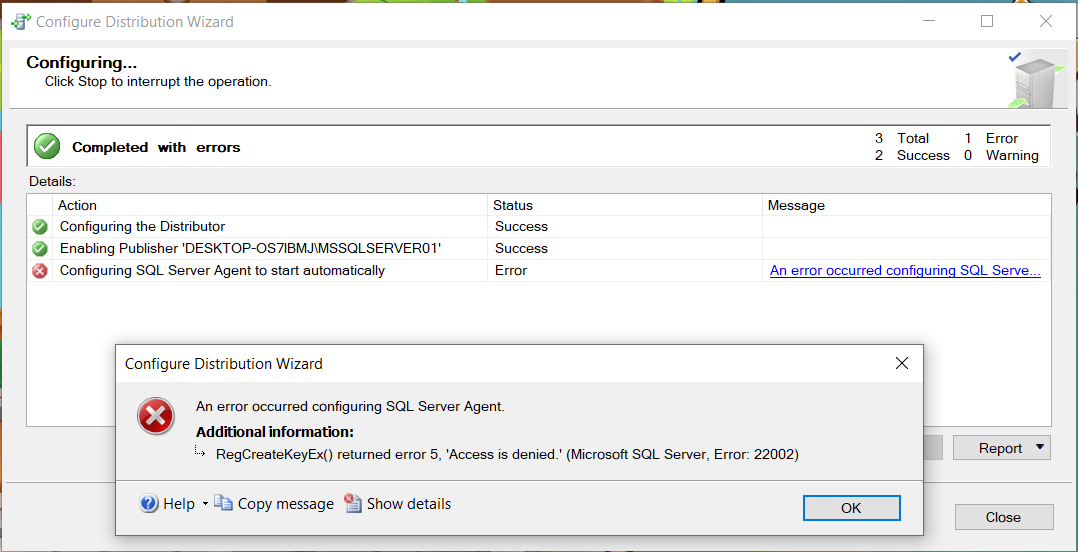
Biz de yaptığımız adımları bu başlıklar üzerinde değerlendirecek olursak.

**Bölüm 1: SQL Server Replikasyonu ile Veritabanı Çoğaltma**

Bu bölüm, SQL Server'ın yerel Replikasyon özelliklerini kullanarak veritabanı çoğaltma sürecinin kurulmasına odaklanmaktadır. Veritabanı replikasyonu, veri ve veritabanı nesnelerini bir kaynak veritabanından bir veya daha fazla hedef veritabanına kopyalamak ve ardından tutarlılığı korumak için bu veritabanları arasında senkronizasyon sağlamak amacıyla kullanılan bir teknolojidir. Bu bölüm, sağlam bir replikasyon topolojisi uygulamak için gerekli olan temel planlama, yapılandırma ve doğrulama aşamalarını detaylandıracaktır. Belirtilen prosedürler, veri yedekliliği sağlamayı, ölçeklenebilir raporlama yeteneklerini kolaylaştırmayı veya verileri uzak ofislere dağıtmayı amaçlayacaktır.

**Adım 1.1: Adım 1: Ortam Hazırlığı ve Distributor Yapılandırması**

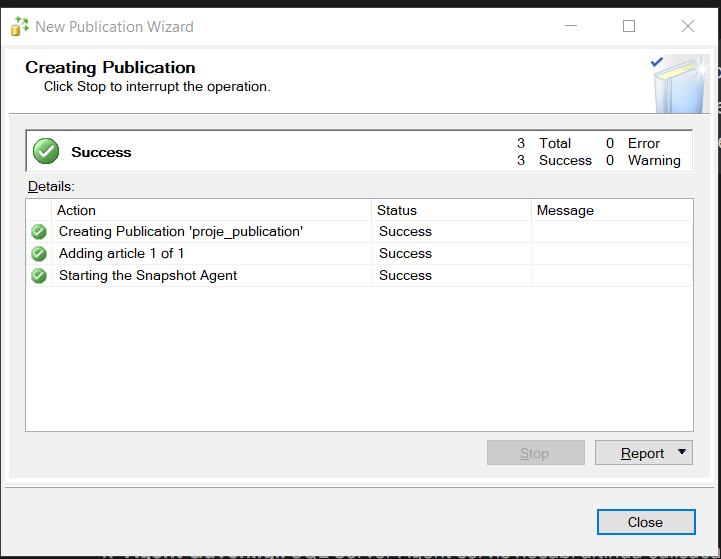
Projenin ilk etabında, Microsoft SQL Server üzerinde veri replikasyon altyapısının tesis edilmesi amacıyla Distributor rolünün konfigürasyonu gerçekleştirilmiştir. Bu kapsamda, Publisher olarak atanan sunucu üzerinde SQL Server Management Studio kullanılarak Configure Distribution Wizard sihirbazı çalıştırılmış; bu işlem neticesinde replikasyon metaverilerinin ve işlem kayıtlarının tutulacağı distribution veritabanı tesis edilmiş ve ilgili sunucunun Yayıncı olarak yetkilendirilmesi sağlanmıştır.



Buradaki söz konusu error durumu son aşama olarak MSSQLSERVER01 için SQL Server Agent’in otomatik olarak başlatılamadığını söylüyor. SQL Server Configuration Manager üzerinden manuel olarak kendi elimizle başlattığımızda bu sorunu ortadan kaldırmamız oldukça basit.

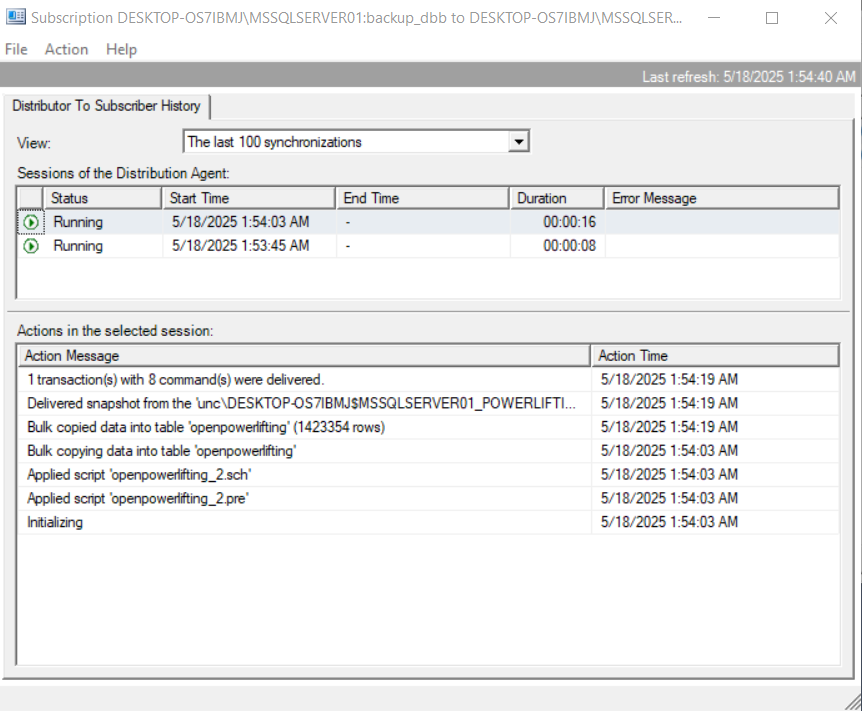
**Adım 1.2: Publication Oluşturma**

Projenin ikinci adımında, veri replikasyonuna kaynak teşkil edecek olan powerlifting\_db isimli veritabanı üzerinde bir yayın tanımlama süreci icra edilmiştir. Bu kapsamda, yayın türü olarak Snapshot Publication seçilmiş ve kaynak veritabanında bulunan tüm tablolar, replike edilecek articles olarak yayına dahil edilmiştir. Snapshot Agent’ın güvenlik yapılandırması, SQL Server Agent servis hesabının yetkileri altında çalışacak şekilde belirlenmiştir. Bu işlemlerin başarıyla tamamlanması neticesinde, powerlifting\_db veritabanındaki veriler, Subscriber sunuculara dağıtılmaya hazır hale getirilmiş ve böylece replikasyon mimarisinin temel bir bileşeni olan yayın başarıyla oluşturulmuştur.



**Adım 1.3: Subscription Oluşturma**

Projenin ilk bölümünün son aşamasında, powerlifting\_db veritabanı için oluşturulan Anlık Görüntü Yayınına (Snapshot Publication) bir abonelik (Subscription) tanımlanmıştır. Bu süreçte, başlangıçta farklı bir sunucu (MSSQLSERVER02) üzerinde hedeflenen abonelik yapılandırmasında anlık görüntü klasörüne erişimle ilgili bir sorunla karşılaşılmış; bu durum, Abone'nin Yayıncı ve Dağıtıcı ile aynı sunucu olan MSSQLSERVER01 üzerine alınması ve backup\_dbb adında yeni bir abonelik veritabanı oluşturulması stratejisiyle çözümlenmiştir. "Push" dağıtım yöntemi ve "sürekli çalışma" (run continuously) senkronizasyon zamanlaması ile yapılandırılan abonelik neticesinde, Dağıtım Agent'ı (Distribution Agent) tarafından kaynak veritabanındaki şema ve yaklaşık 1.4 milyon satırlık veri, backup\_dbb hedef veritabanına başarıyla aktarılmış, bu durum hem Replication Monitor üzerinden hem de kullanıcı tarafından hedef veritabanı içeriğinin doğrudan incelenmesiyle teyit edilmiştir. Böylelikle, SQL Server Replication kullanılarak veri çoğaltma ve temel senkronizasyonun sağlanması hedefi başarıyla yerine getirilmiş ve projenin ilk ana bölümü tamamlanmıştır.

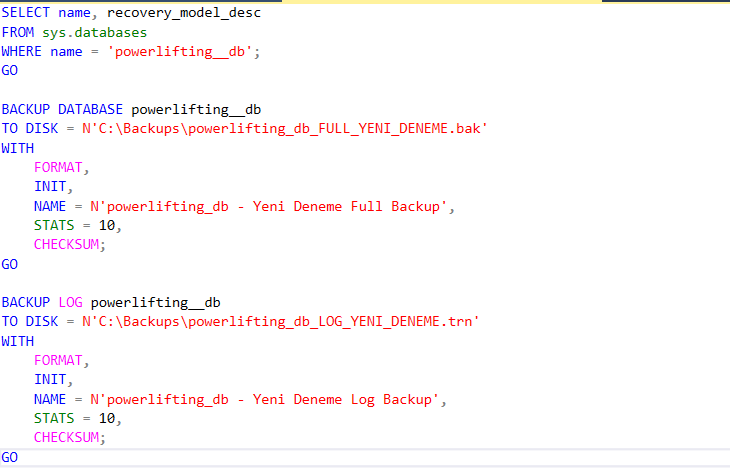


**Bölüm 2: Yük Dengeleme**

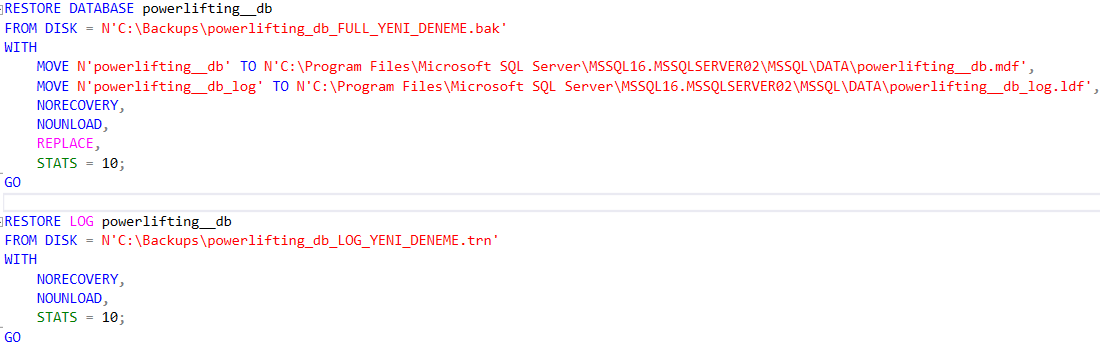
Projenin ikinci bölümü, Microsoft SQL Server ortamında öncelikli olarak yüksek erişilebilirlik) sağlamak amacıyla, öğrenme ve karşılaştırma hedefleri doğrultusunda Database Mirroring teknolojisinin incelenmesi ve yapılandırılmasına odaklanacaktır. Bu kapsamda, Principal ve Mirror SQL Server örnekleri üzerinde özel yansıtma uç noktalarının Endpoints tanımlanması, yansıtılacak veritabanının Full Recovery Model’ine getirilip ilk tam yedeğinin alınarak ayna sunucusuna NORECOVERY seçeneğiyle geri yüklenmesi ve ardından senkron veya asenkron çalışma modlarından biri seçilerek yansıtma oturumunun başlatılması adımları uygulamalı olarak gerçekleştirilecektir. Ayrıca, automatic failover senaryoları için isteğe bağlı bir Witness sunucu yapılandırması da değerlendirilecektir. Yük dengeleme hedefi için ise, doğrudan sorgulanamayan ayna veritabanı üzerinde Database Snapshots oluşturularak bu statik kopyalar üzerinden okuma işlemleri yapılması ve bu yaklaşımın getirdiği avantaj ve kısıtlamalar incelenecektir. Bu bölümün temel amacı, veritabanı servis sürekliliğini artırmak için alternatif bir yüksek erişilebilirlik çözümünü deneyimlemek ve yük dengeleme kavramını bu teknolojinin sunduğu dolaylı yöntemler çerçevesinde ele almaktır.

**Adım 2.1: Ön Koşullar ve Ortam Hazırlığı**

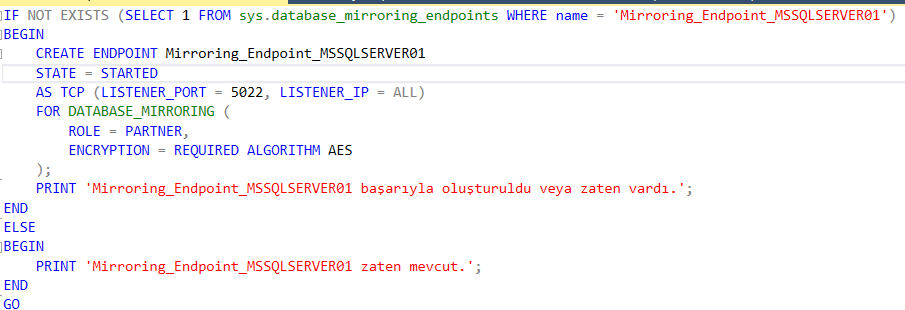
Database Mirroring oturumunu başlatmadan önce, Principal sunucu olarak görev yapacak MSSQLSERVER01 SQL Server örneği üzerindeki powerlifting\_db veritabanının yansıtılmaya uygun hale getirilmesi amacıyla gerekli yedekleme işlemleri gerçekleştirilmiştir. Bu kapsamda, veritabanının kurtarma modelinin FULL olduğu teyit edildikten sonra, T-SQL komutları kullanılarak öncelikle veritabanının güncel bir full backup alınmıştır. Tam yedekleme işleminin başarıyla tamamlanmasının hemen ardından, yansıtma için kesintisiz bir günlük zinciri oluşturmak amacıyla aynı veritabanının bir transaction log backup alınmıştır.

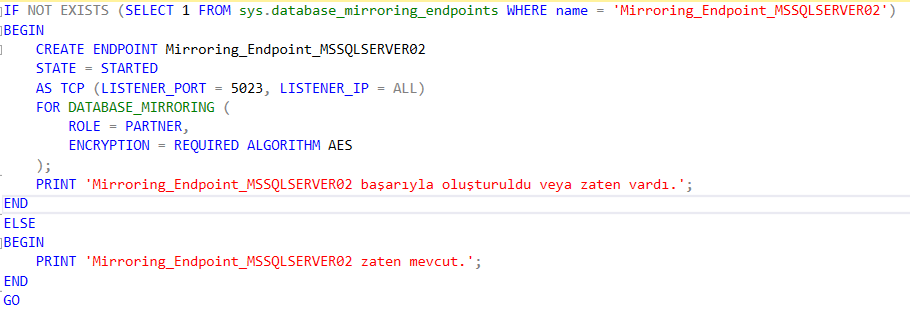


Birincil sunucuda powerlifting\_db veritabanı için gerekli yedekleme işlemleri tamamlandıktan sonra, Mirror sunucu olarak görev yapacak MSSQLSERVER02 SQL Server örneği üzerinde bu veritabanı yansıtma için özenle hazırlanmıştır. Bu amaçla, MSSQLSERVER01 üzerinden alınan tam yedek dosyası, MSSQLSERVER02'nin ilgili veri ve günlük dosyası yolları belirtilerek ve veritabanını daha sonraki işlem günlüğü yedeklerini kabul edebilir durumda bırakmak için T-SQL komutları aracılığıyla WITH NORECOVERY seçeneğiyle geri yüklenmiştir. Tam yedek geri yüklemesinin başarıyla tamamlanmasını takiben, yine MSSQLSERVER01'den alınan işlem günlüğü yedeği de aynı şekilde WITH NORECOVERY seçeneğiyle MSSQLSERVER02'deki powerlifting\_db veritabanı üzerine geri yüklenmiştir. Bu işlemler neticesinde, powerlifting\_db veritabanı MSSQLSERVER02 üzerinde Restoring durumuna getirilerek yansıtma oturumunun başlatılması için temel zemin oluşturulmuştur.



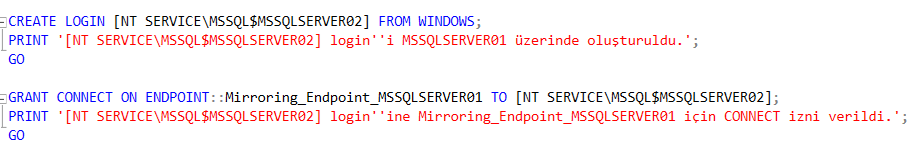
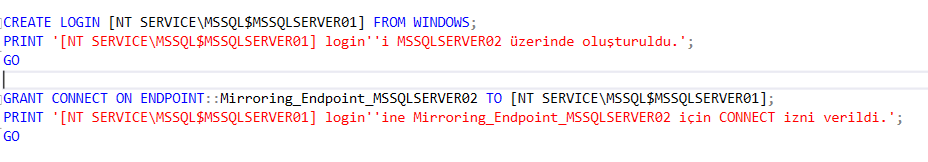
Birincil powerlifting\_db veritabanının yedeklenmesi ve bu yedeklerin Ayna sunucudaki powerlifting\_db veritabanına Restoring durumunda başarıyla geri yüklenmesiyle her iki veritabanı da yansıtma için hazır hale getirilmiştir. Bu hazırlıkların ardından, Principal sunucu olarak görev yapacak MSSQLSERVER01 ile Mirror sunucu MSSQLSERVER02 arasında yansıtma trafiği için güvenli ve özel bir iletişim kanalı tesis etmek amacıyla Endpoints yapılandırılmıştır. Her iki SQL Server örneği üzerinde de T-SQL komutları aracılığıyla, varsayılan olarak TCP port 5022 üzerinden dinleme yapacak şekilde, PARTNER rolüyle ve AES algoritması kullanılarak şifrelenmiş DATABASE\_MIRRORING için özelleşmiş endpoint'ler oluşturulmuş ve STARTED durumuna getirilmiştir. Yapılan kontrol sorguları, her iki endpoint'in de aktif olduğunu ve yansıtma oturumuna katılmak üzere doğru rol ve port ayarlarıyla çalıştığını teyit etmiştir. Bu yapılandırma, Birincil ve Ayna sunucuların yansıtma verilerini güvenli bir şekilde senkronize edebilmesi için gerekli olan temel iletişim altyapısını tamamlamıştır.



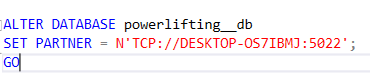


**Adım 2.2: Veritabanı Yansıtma Oturumunu Başlatma**

Birincil ve Ayna sunucular arasında güvenli endpoint iletişimi tesis edildikten sonra, her bir SQL Server örneğinin altında çalışan servis hesaplarının, karşıdaki sunucunun yansıtma endpoint'ine bağlanabilmesi için gerekli kimlik doğrulama ve yetkilendirme adımları atılmıştır. Bu doğrultuda, MSSQLSERVER01 SQL Server örneği üzerinde, MSSQLSERVER02 örneğinin servis hesabı için bir Windows kimlik doğrulamalı SQL Server oturum açma hesabı oluşturulmuştur. Benzer şekilde, MSSQLSERVER02 SQL Server örneği üzerinde de MSSQLSERVER01 örneğinin servis için bir login tanımlanmıştır. Bu login'ler oluşturulduktan sonra, T-SQL komutları aracılığıyla her bir sunucudaki yansıtma endpoint'i üzerinde, karşı sunucunun ilgili servis hesabı login'ine CONNECT izni verilmiştir. Bu işlemler neticesinde, her iki SQL Server örneğinin servis hesapları, yansıtma oturumunu kurmak ve sürdürmek için birbirlerinin endpoint'lerine güvenli bir şekilde erişim yetkisine sahip olmuştur.



Tüm ön hazırlıklar tamamlandıktan sonra, powerlifting\_db veritabanı için yansıtma oturumu Principal sunucu olan MSSQLSERVER01 üzerinden başlatılmıştır. Bu amaçla, MSSQLSERVER01'e, Mirror sunucusunun MSSQLSERVER02 örneğindeki 5022 numaralı portta dinleyen endpoint olduğu bildirilmiştir. Bu komutun başarıyla icra edilmesiyle birlikte, MSSQLSERVER01 ve MSSQLSERVER02 arasında powerlifting\_db veritabanı için bir yansıtma ortaklığı kurulmuş ve Yüksek Güvenlik Modunda senkronizasyon başlamıştır. Sonuç olarak, MSSQLSERVER01 üzerindeki powerlifting\_db veritabanı Principal, Synchronized durumuna, MSSQLSERVER02 üzerindeki powerlifting\_db veritabanı ise Mirror, Synchronized / Restoring durumuna geçerek, veritabanı için yüksek erişilebilirlik altyapısının ilk adımı başarıyla tesis edilmiştir. Bu aşamada, Birincil veritabanında yapılan tüm değişiklikler anlık olarak Ayna veritabanına yansıtılmaya başlanmıştır.

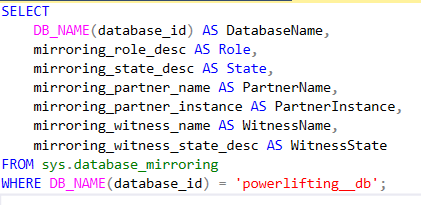


**Bölüm 3: Failover Senaryoları**

Projenin üçüncü ve son bölümü, önceki aşamada powerlifting\_\_db veritabanı için Microsoft SQL Server üzerinde MSSQLSERVER01 Principal ve MSSQLSERVER02 Mirror sunucuları arasında başarıyla tesis edilen Database Mirroring yapılandırması temel alınarak, sistemin sürekliliğini ve veri bütünlüğünü güvence altına almak amacıyla çeşitli failover stratejilerinin uygulanmasına ve yönetimsel işlemlerin gerçekleştirilmesine odaklanacaktır. Bu bölümde, planlı bakım veya kontrollü rol değişimi senaryoları için T-SQL komutları kullanılarak Manual Failover işleminin nasıl yapılacağı ve sunucu rollerinin nasıl değiştirileceği adım adım ele alınacaktır. Ayrıca, Principal sunucunun beklenmedik bir şekilde hizmet dışı kalması durumunda, Mirror sunucusunun Forced Service yöntemiyle, potansiyel veri kaybı riskleri de dikkate alınarak, servisi nasıl devralacağı ve veritabanının nasıl erişilebilir hale getirileceği incelenecektir. Ek olarak, sistemin kesintilere karşı otonom bir yanıt verme kapasitesini artırmak amacıyla bir Witness sunucunun yapılandırılması ve Database Mirroring oturumunun sonlandırılması gibi kritik yönetimsel işlemler de T-SQL düzeyinde detaylandırılacaktır.

**Adım 3.1: Mevcut Veritabanı Yansıtma Durumunun Teyit Edilmesi**

Database Mirroring oturumunun sağlıklı bir şekilde kurulduğunu ve powerlifting\_\_db veritabanı için rollerin beklendiği gibi yapılandırıldığını doğrulamak amacıyla, hem Principal hem de Mirror sunucu örnekleri üzerinde sys.database\_mirroring sistem kataloğu görünümüne yönelik bir T-SQL sorgusu çalıştırılmıştır. Bu sorgu, yansıtma yapılandırmasındaki her bir veritabanının rolünü, durumunu ve ortak bilgilerini detaylı bir şekilde raporlamaktadır.



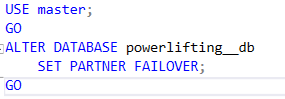
Sorgunun çıktısı aşağıda gibidir. Doğru sonuçlar alındığı görülmektedir.





**Adım 3.2: Manual Failover İşleminin Gerçekleştirilmesi**

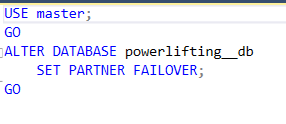
Database Mirroring oturumunun sağlıklı ve SYNCHRONIZED olduğu Adım 3.1'de teyit edildikten sonra, planlı bir rol değişimi senaryosu uygulamak amacıyla Manuel Failover işlemi gerçekleştirilmiştir. Bu işlem, mevcut Principal sunucu olan MSSQLSERVER01 üzerinde, powerlifting\_\_db veritabanı için ALTER DATABASE powerlifting\_\_db SET PARTNER FAILOVER; T-SQL komutunun yürütülmesiyle başlatılmıştır. Komutun başarıyla tamamlanması neticesinde, MSSQLSERVER01 sunucusu Mirror rolünü üstlenirken, MSSQLSERVER02 sunucusu Principal rolünü alarak powerlifting\_\_db veritabanına hizmet vermeye başlamıştır. Bu rol değişimi sırasında herhangi bir veri kaybı yaşanmamış ve veritabanı erişilebilirliği kesintisiz bir şekilde devam etmiştir. İşlem sonrasında sys.database\_mirroring üzerinden yapılan durum kontrolleri, rollerin beklendiği gibi değiştiğini ve oturumun yeni rollerle SYNCHRONIZED durumda sağlıklı bir şekilde devam ettiğini doğrulamıştır.





**Adım 3.3: Rolleri Geri Değiştirme**

Önceki Manuel Failover işlemi neticesinde MSSQLSERVER02 sunucusu Principal, MSSQLSERVER01 sunucusu ise Mirror rolünü üstlenmişti. Bu adımda, sistemin esnekliğini ve rollerin sorunsuz bir şekilde geri çevrilebildiğini göstermek amacıyla, rollerin tekrar eski haline getirilmesi için ikinci bir Manuel Failover işlemi gerçekleştirilmiştir. Bu işlem, mevcut Principal sunucu olan MSSQLSERVER02 üzerinde, powerlifting\_\_db veritabanı için ALTER DATABASE powerlifting\_\_db SET PARTNER FAILOVER; T-SQL komutunun yürütülmesiyle başlatılmıştır. Komutun başarıyla tamamlanması sonucunda, MSSQLSERVER02 sunucusu tekrar Mirror rolünü üstlenirken, MSSQLSERVER01 sunucusu orijinal Principal rolüne geri dönmüştür. Bu rol değişimi sırasında da herhangi bir veri kaybı yaşanmamış ve veritabanı erişilebilirliği kesintisiz devam etmiştir. Yapılan durum kontrolleri, rollerin beklendiği gibi değiştiğini ve oturumun yeni rollerle SYNCHRONIZED durumda sağlıklı bir şekilde devam ettiğini doğrulamıştır.

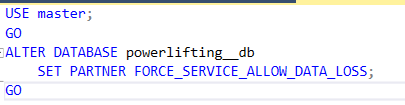


Sorgunun çıktısı aşağıda gibidir. Doğru sonuçlar alındığı görülmektedir.



**Adım 3.4: Zorlanmış Servis ile Yük Devretme**

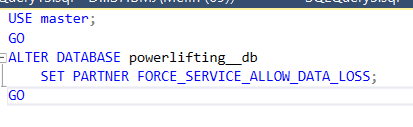
Veritabanı yansıtma yapılandırmasının kritik bir testi olarak, Birincil sunucu MSSQLSERVER01'in beklenmedik bir şekilde hizmet dışı kaldığı bir felaket senaryosu canlandırılmıştır. Bu durumda, powerlifting\_\_db veritabanına erişimin hızla yeniden sağlanması amacıyla, Ayna sunucu olan MSSQLSERVER02 üzerinde ALTER DATABASE powerlifting\_\_db SET PARTNER FORCE\_SERVICE\_ALLOW\_DATA\_LOSS; T-SQL komutu yürütülerek Zorlanmış Servis başlatılmıştır. Bu komutun icrası neticesinde, MSSQLSERVER02 sunucusu Birincil rolünü üstlenmiş ve powerlifting\_\_db veritabanı, potansiyel veri kaybı riski kabul edilerek istemci bağlantılarına açılmıştır. Bu işlem, Birincil sunucunun tamamen kullanılamaz hale geldiği durumlarda servis kesintisini en aza indirmek için kritik bir müdahale yöntemi olarak değerlendirilmiştir. Eski Birincil sunucu tekrar çevrimiçi olduğunda, yansıtma oturumunun durumu ve veritabanlarının senkronizasyonu için ek adımlar gerekeceği gözlemlenmiştir.





**Adım 3.4: Zorlanmış Servis ile Yük Devretme**

Veritabanı yansıtma yapılandırmasının süreklilik ve dayanıklılık testleri kapsamında, Birincil sunucu MSSQLSERVER01'in beklenmedik bir şekilde hizmet dışı kaldığı ve ulaşılamaz olduğu kritik bir felaket senaryosu canlandırılmıştır. Bu durumda, powerlifting\_\_db veritabanına erişimin en kısa sürede ve minimum kesintiyle yeniden sağlanması amacıyla, Ayna sunucu olan MSSQLSERVER02 üzerinde Zorlanmış Servis başlatma prosedürü uygulanmıştır. Bu işlem, ALTER DATABASE powerlifting\_\_db SET PARTNER FORCE\_SERVICE\_ALLOW\_DATA\_LOSS; T-SQL komutunun MSSQLSERVER02 üzerinde yürütülmesiyle gerçekleştirilmiştir. Komutun başarıyla icra edilmesi neticesinde, MSSQLSERVER02 sunucusu Birincil rolünü üstlenmiş ve powerlifting\_\_db veritabanı, potansiyel veri kaybı riski kabul edilerek istemci bağlantılarına açılmıştır. Bu müdahale, Birincil sunucunun tamamen kullanılamaz hale geldiği durumlarda servis devamlılığını sağlamak için hayati bir strateji olarak değerlendirilmiştir. Eski Birincil sunucu olan MSSQLSERVER01'in tekrar çevrimiçi olması durumunda, yansıtma oturumunun durumu ve veritabanlarının yeniden senkronizasyonu için ek manuel adımlar gerekeceği tespit edilmiştir.

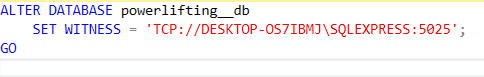


Kontrol sorgusu çalıştırıldığında



**Adım 3.5: Witness Sunucu Eklemek ve Otomatik Yük Devretmeyi Yapılandırmak**

Mevcut powerlifting\_\_db Database Mirroring oturumunun High Safety with Automatic Failover yeteneği kazanması amacıyla, sisteme bir Witness sunucu dahil edilmiştir. Tanık sunucu olarak ana bilgisayar üzerinde çalışan SQLEXPRESS SQL Server örneği kullanılmıştır. Bu işlem öncesinde, SQLEXPRESS örneği üzerinde DATABASE\_MIRRORING için Endpoint TCP port 5025 üzerinden WITNESS rolüyle oluşturulmuş ve SQL Server servis hesaplarına gerekli bağlantı izinleri verilmiştir. Ardından, mevcut Principal sunucu olan MSSQLSERVER01 üzerinden, powerlifting\_\_db veritabanı için aşağıdaki T-SQL komutu yürütülerek Tanık sunucu yansıtma oturumuna başarıyla atanmıştır:



Bu soru sonrası kontrol işlemi yapıldığında aşağıdaki sonuçları alırız.



