**Proje 3: Veritabanı Güvenliği ve Erişim Kontrolü**

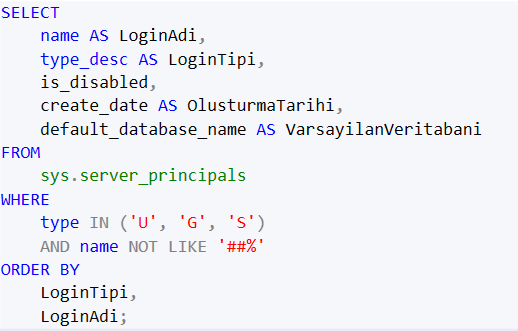
**Kullanılan veri tabanı:** Wide World Importers sample database v1.0

**Link:** <https://github.com/Microsoft/sql-server-samples/releases/tag/wide-world-importers-v1.0>

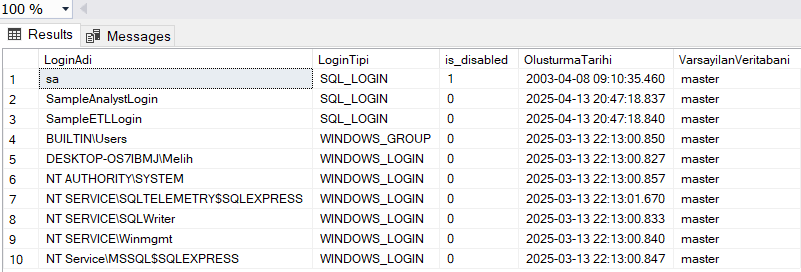
Bu projede istenen adımlar 4 temel başlık altında değerlendirilmiş. 1- Erişim Yönetimi: 2-Veri Şifreleme 3-SQL Injection Testleri 4-Audit Logları 5-Biz de yaptığımız adımları bu başlıklar üzerinde değerlendirecek olursak.

**Erişim Yönetimi**

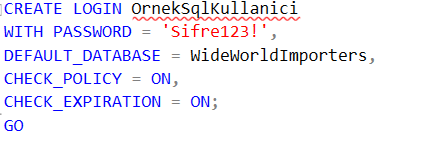
Mevcut Sunucu Girişlerini İnceleme işlemi ile başlayabiliriz. Aşağıdaki sorgu, sunucu seviyesindeki girişleri ve türlerini listeleyecektir.

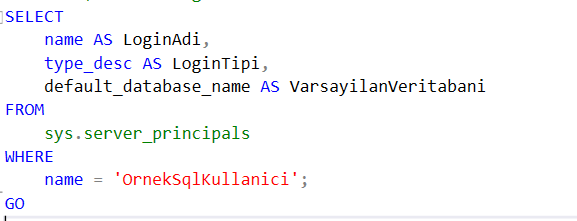


Bu sorgunun sonucunu aşağıda görüntüleyebiliriz. beklendiği gibi hem SQL\_LOGIN hem de WINDOWS\_LOGIN, kullanıcının kendi hesabı ve çeşitli NT servis hesabı tiplerinde girişlerin bulunduğunu gösteriyor.

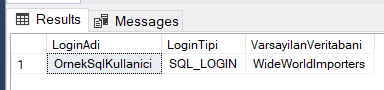


Şimdi SQL Server Authentication kullanarak yeni bir sunucu girişi (login) oluşturalım. Aşğıdaki sorguyu çalıştırarak bu işlemi sağlayabiliriz.

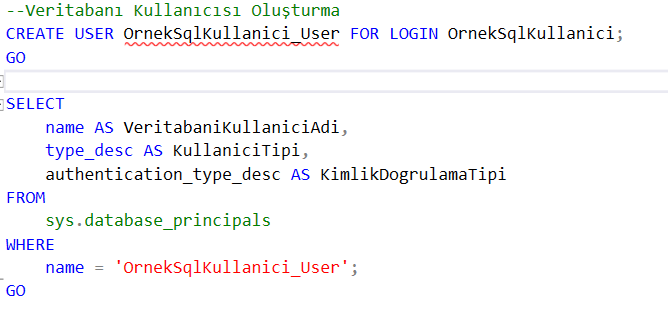


Aşağıdaki sorguyu çalıştırarak login oluşumu başarılı olup olmadığını kontrol edelim.

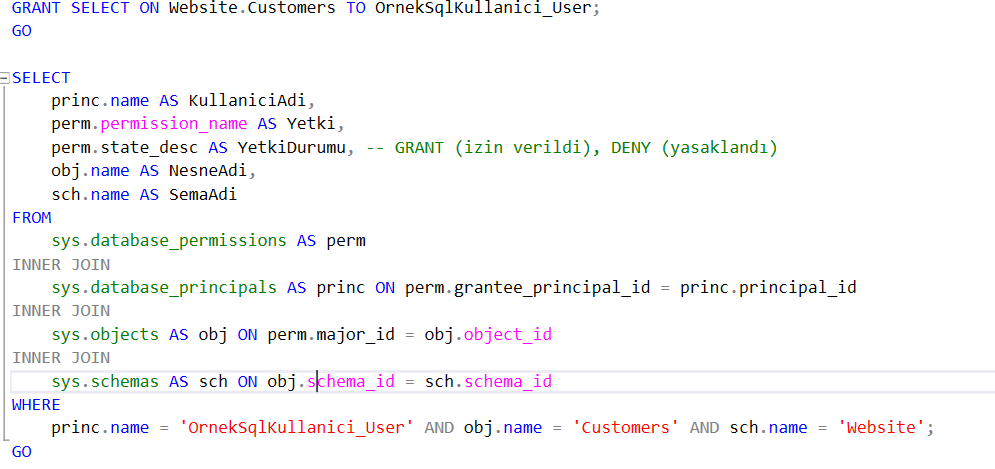
Bu sorgu sonucunda aşağıdaki görüldüğü gibi işlemin başarılı olduğunu görüyoruz.



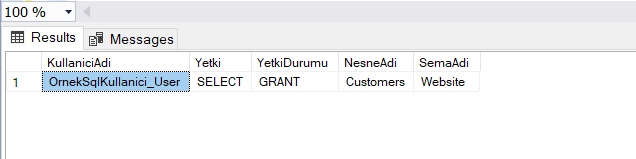
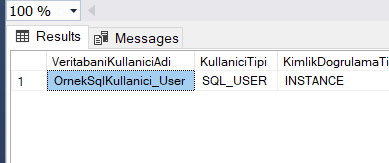
Aşağıdaki sorguyu çalıştırarak bir veritabanı kullanıcısı oluşturabiliriz.



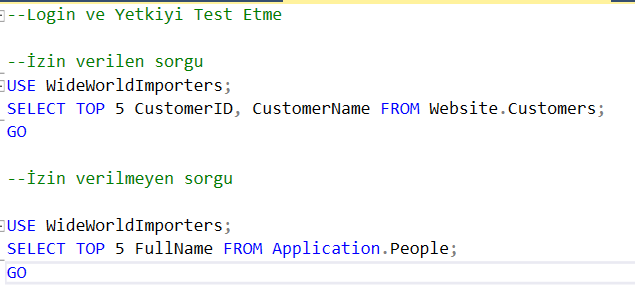
Aşağıdaki sorguyu çalıştırarak veritabanı kullanıcımıza yetki verebiliriz. Bu sayede hangi kısımlara erişip erişemeyeceğini belirlemiş oluruz.



Bu sorguları çalıştırdığımızda aşağıdaki sonuca ulaşırız. Bu da istenen yetkilerin kullanıcıya sağladığını gösterir.

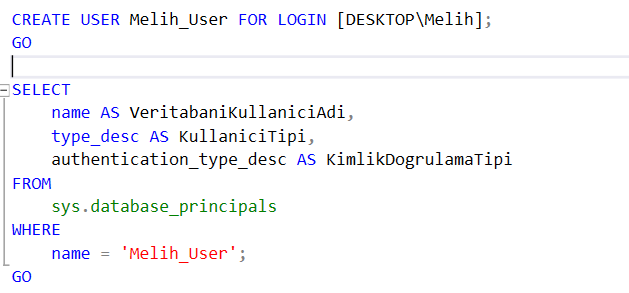


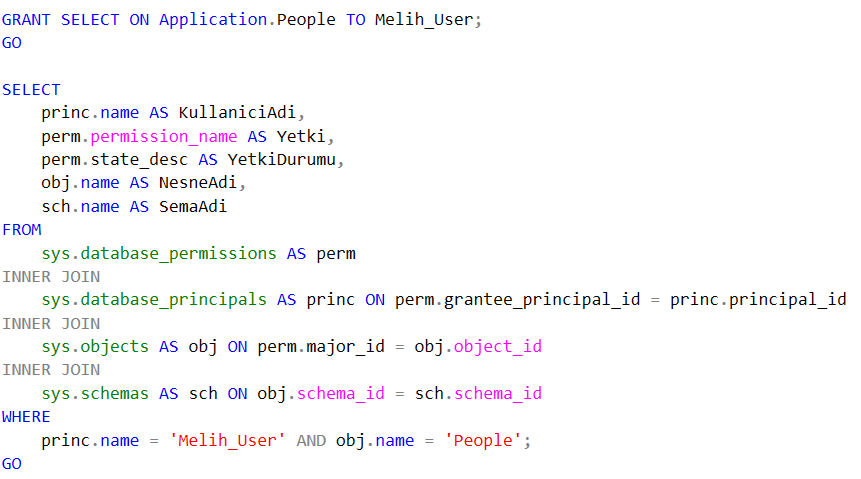
Verdiğimiz yetkileri doğru şekilde kullanabiliyor muyuz bunu denemek için sorgu çalıştıralım. Aşağıdaki sorguları çalıştırdığımızda ilk sorgunun çalıştığını fakat ikinci sorgunun hata verdiğini görürüz



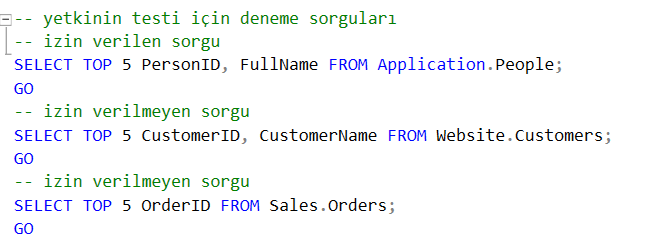
Bu adımları gerçekleştirdikten sonra Windows login için de bir kullanıcı oluşturalım ve erişimlerini test edelim.

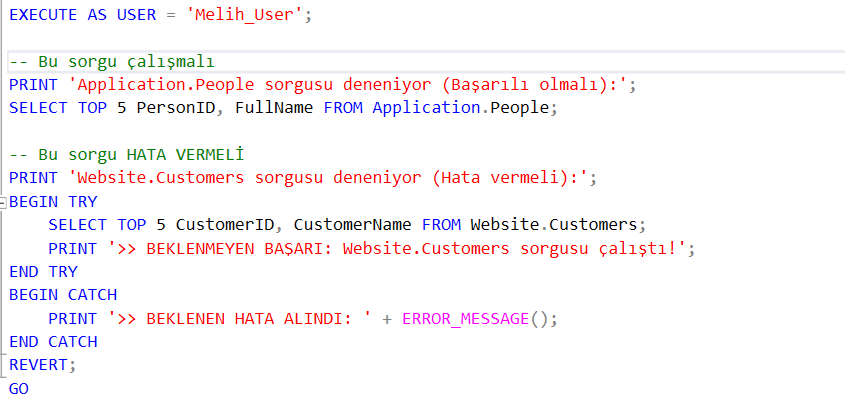
DESKTOP kullanıcısı için login'i için WideWorldImporters veritabanı içinde bir kullanıcı oluşturalım.

Aşağıdaki sorgu ile kullanıcımıza yetki verelim.



Aşağıdaki sorguyu çalıştırarak izinleri test edebiliriz. Fakat bu sorgular doğru çalışmayacaktır. Çünkü biz admin pc üzerinden çalıştırdığımız için mevcut kısıtlamalar işe yaramaz ve istediğimiz yere erişebiliriz.

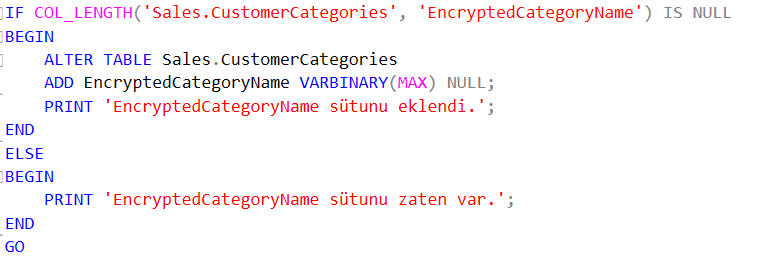


Bu yüzden başka bir kullanıcıymış gibi davranarak test etmeliyiz. Aşağıdaki sorgu bunu gerçekleştirir. Tam anlamıyla kullanıcıymış gibi davranarak sorguları çalıştırır. Bu sayede kısıtlamalara tabii oluruz.

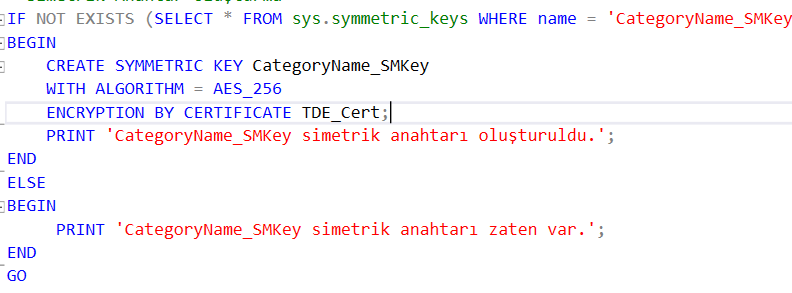
**Veri şifreleme**

Veri şifrelemenin temel amacı, veritabanında saklanan hassas bilgilerin yetkisiz erişime karşı korunmasıdır. Özellikle veritabanı dosyalarının (.mdf, .ldf) veya yedeklerin fiziksel olarak ele geçirilmesi durumunda, verilerin okunamaz halde olmasını sağlamayı hedefler (at-rest encryption).

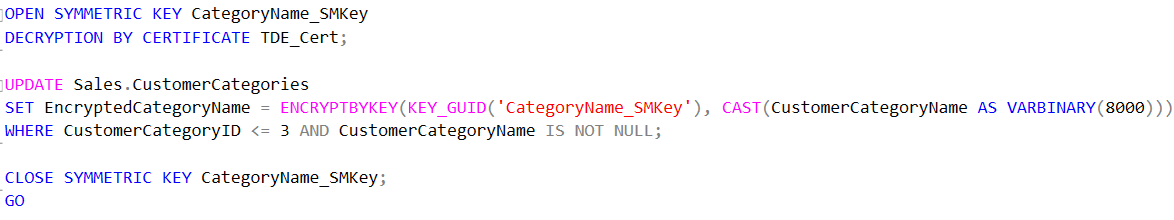
Aşağıdaki sorguyu çalıştırarak Sales.CustomerCategories tablosuna, şifrelenmiş kategori adlarını binary formatta saklamak üzere EncryptedCategoryName adında yeni bir sütun ekledik.



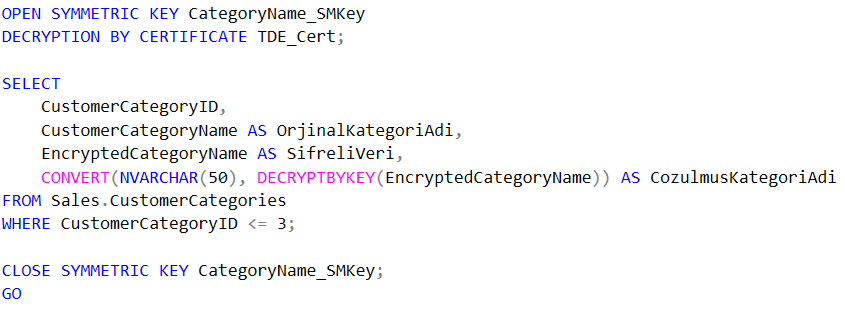
Aşağıdaki sorgu ile CategoryName\_SMKey adında yeni bir simetrik şifreleme anahtarı oluşturduk. Şifreleme algoritması olarak AES\_256'yı seçtik. Bu yeni anahtarın kendisini de güvenli bir şekilde saklamak için, veritabanında daha önceden var olan TDE\_Cert isimli sertifikayı kullanarak şifreledik.



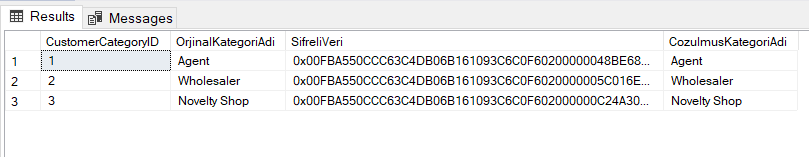
Aşağıdaki sorgu ile. İlk 3 kategoriye ait EncryptedCategoryName sütunları, orijinal kategori adlarının şifrelenmiş binary halleriyle güncellendi. Bu sayede örnek bir şifreleme gerçekleştirmiş olduk.



Bu sayede şifreleme işlemlerimizi tamamlamış olduk ve son bir sorgu ile istediğimiz veriler gerçekten şifrelenmiş mi diye kontrol edebiliriz. Aşağıdaki sorgunun sonucunu inceleyelim.

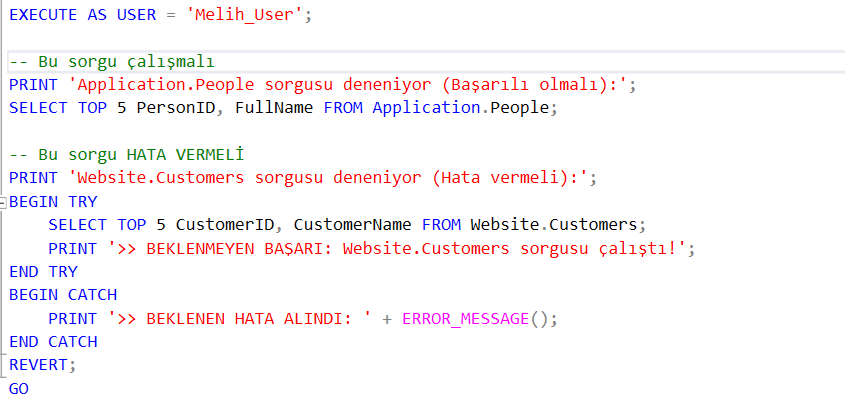


Aşağıdaki görselde de görüldüğü gibi şifrelemiş olduğumuz sütunumuzun hem şifreli verisi hem de çözülmüş kategorisi doğru şekilde görünmekte. Bu da şifreleme işlemimizin başarıyla gerçekleştiğini gösteriyor.

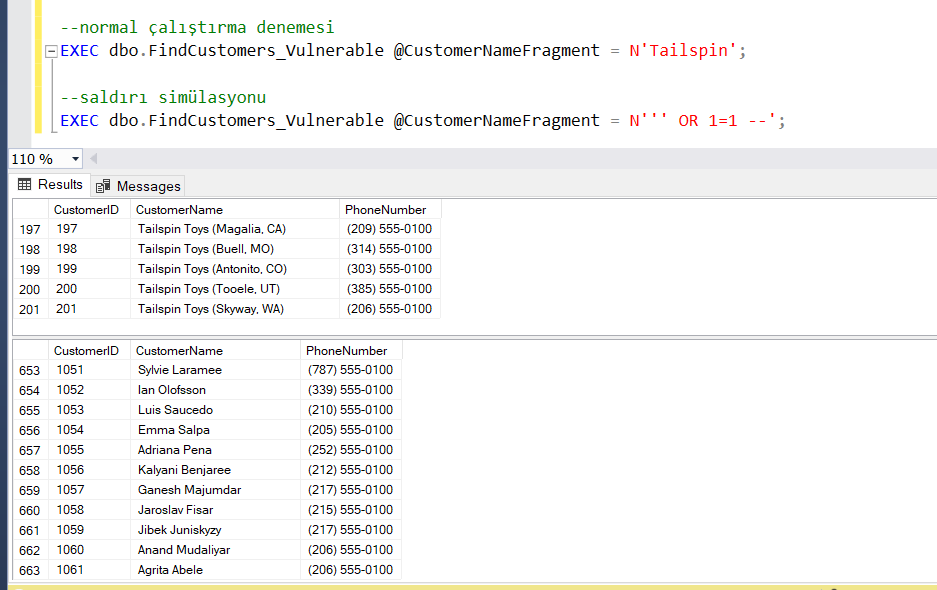


**SQL Injection Testleri**

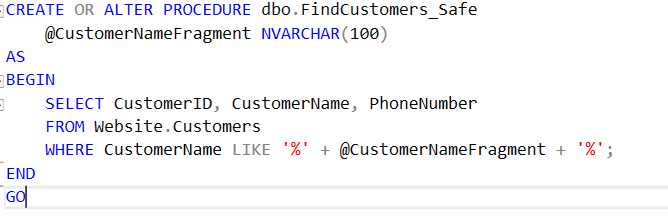
İlk olarak bilerek güvenlik açıkları içeren bir soru yazarak çalıştırıyoruz.



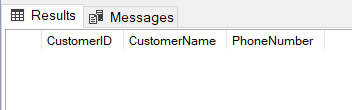
Şimdi bunun üzerine bir saldırı denemesi yapıyoruz. İlik olarak normal şekilde bir sorgulama işlemi yaparak saldırı denemesiyle olan farkını karşılaştıracağız. İkinci sorguda daha fazla satır olmasını bekliyoruz çünkü bir güvenlik zafiyeti verdik ve bu zafiyetten faydalanarak erişmememiz gereken verilere erişebileceklerini test edeceğiz.



Bu durumu çözmek için güvenli prosedür yani parametreli sorgu oluşturulabilir. Aşağıdaki sorgu ile bu işlemi gerçekleştiriyoruz.



Tekrardan güvenli ve güvensiz şekilde erişme testlerini yaptığımızda artık güvensiz olan sorgunun bir sonuç döndürmediğini görüyoruz. Bir önceki sorguda yaptığımız işlem başarıyla gerçekleşmiş.



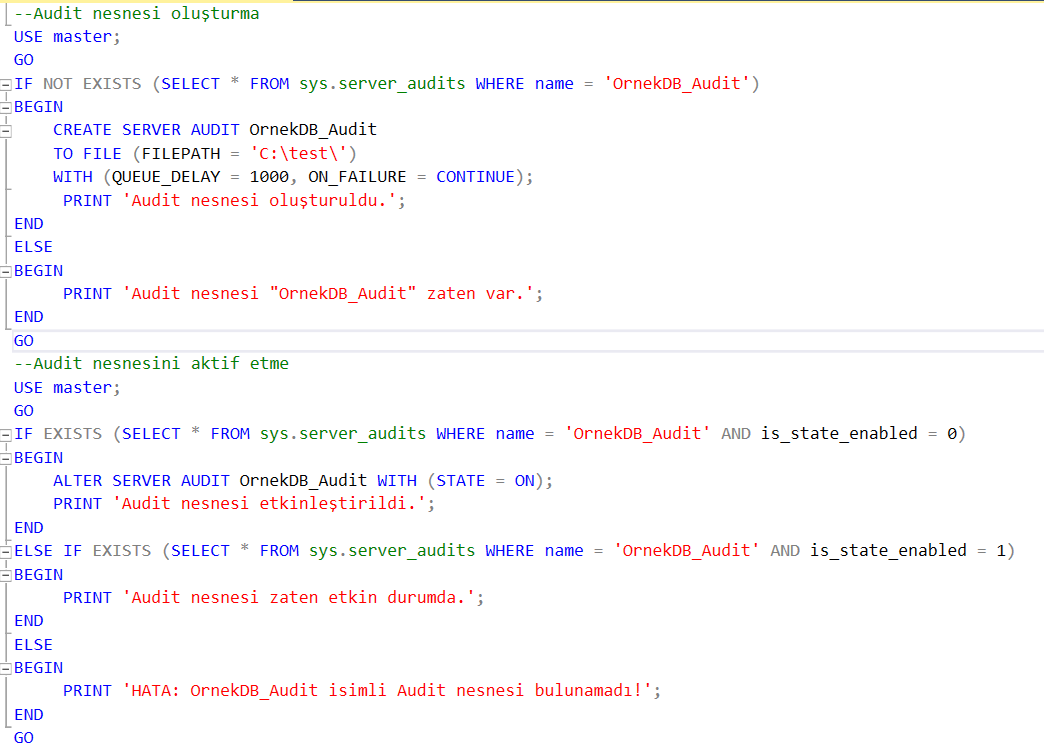
**Audit Logları**

SQL Server Audit, sunucu ve veritabanı seviyesinde gerçekleşen belirli olayları (eylemleri) izlemenizi ve kaydetmenizi sağlayan güçlü bir özelliktir. Bu özellik sayesinde kim, ne zaman, ne yaptı? sorularının cevaplarını bulabilirsiniz.

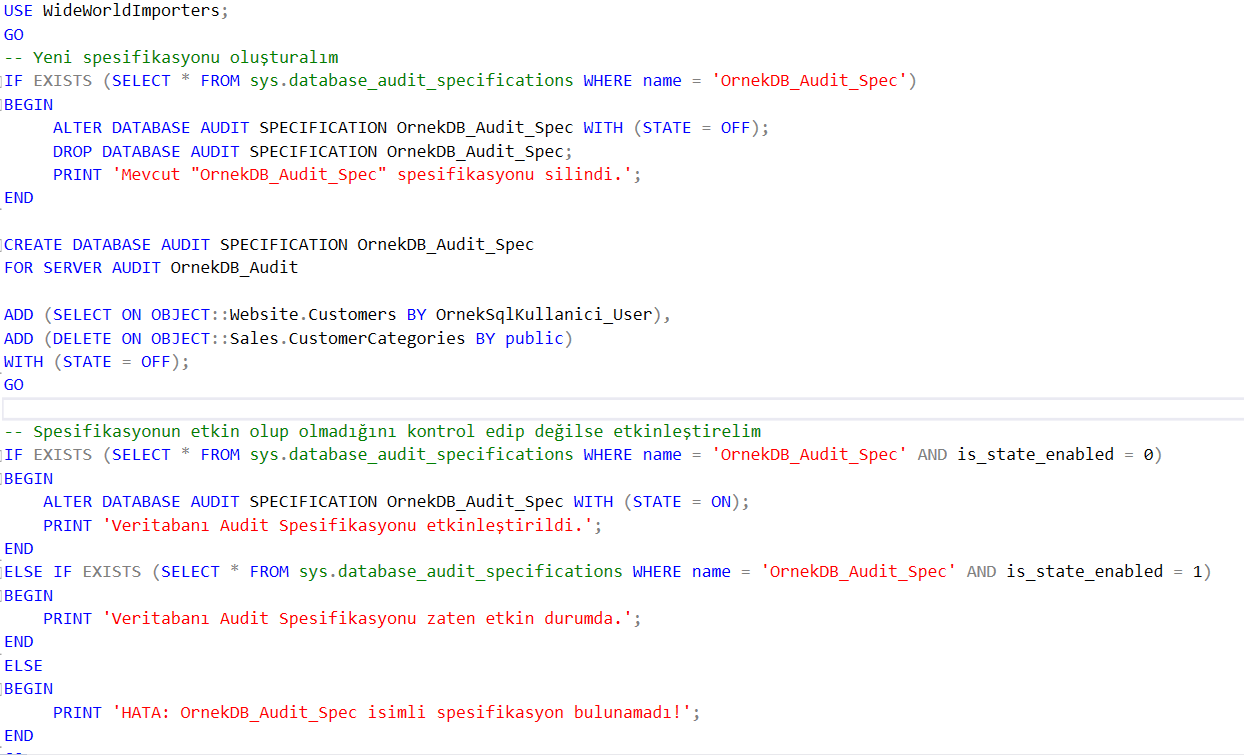
İlk adım, denetim kayıtlarımızın nereye yazılacağını belirleyen ana Audit nesnesini oluşturmaktır. Kayıtları genellikle bir dosyaya yazmak en esnek yöntemdir.

Aşağıdaki iki sorguyu çalıştırarak bir audit nesnesi oluşturmak mümkün. Bunun için ilk önce, arka planda çalışan SQL Server Veritabanı Altyapısı Hizmetini çalıştıran Windows kullanıcı hesabına audit işlemleri için yetki vermek gerekiyor. Bizim yetkili olmamız yeterli olmuyor. Kaydedeceği klasörü oluşturup okuma yazma ve hatta değiştirme yetkisi vermek gerekiyor.

Bu işlemleri yaptıktan sonra aşağıdaki iki sorguyu çalıştırarak audit nesnesi oluşturulabilir.

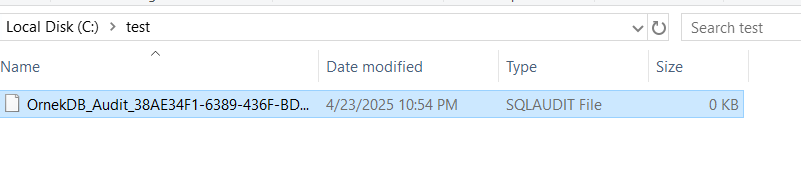


Daha sonra mevcut audit üzerinde bir spesifikasyon oluşturarak aktif edebiliriz. Aşağıdaki iki sorgu bunu sağlar.



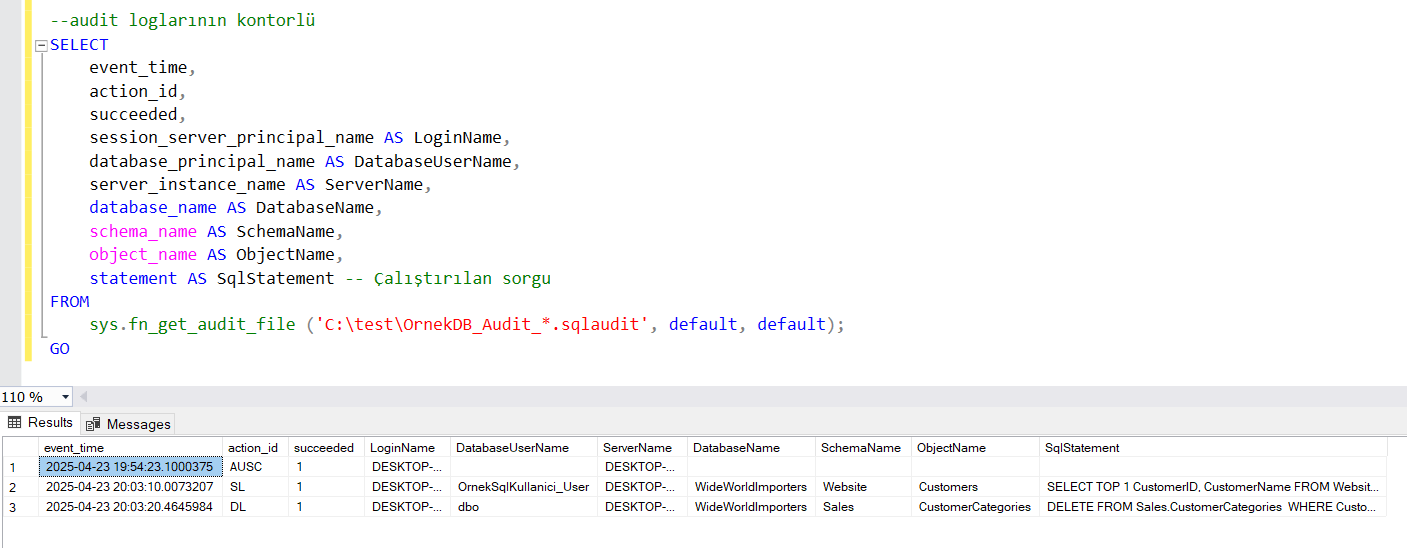
Bu sorgunun sonucunda aktifleştirildi bildirimini alarak işlemleri başarıyla tamamladığımızı anlayabiliriz. Son olarak Log kaydı oluşması için örnek bir SELECT ve DELETE sorgusu çalıştıralım. Bu işlemi yaptığımızda audit nesnesinin bağlı olduğu ilgili dosyaya log kaydedilecektir.





Görüldüğü üzere deneme sorgularımızı çalıştırdıktan sonra log kaydımız oluşmuş. Son olarak içeriğini görüntüleyerek sağlamasını yapalım.

Aşağıda görüldüğü gibi sorguyu çalıştırdığımızda log sonuçlarımızı ve nelerin kayıt edildiğini gözlemleyebiliyoruz.



Bu sayede audit log oluşturma ve gözlemleme işlemini de başarıyla tamamladık.