

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



Ağ Tabanlı Paralel Dağıtım Sistemleri
(BLM4522)

Sümeyye TEKİN – 20290296
(Github: github.com/sumeyye4/BLM4522)

VERİTABANI YEDEKLEME VE FELAKETTEN KURTARMA PLANI

AdventureWorks2022 veritabanı için yedekleme ve felaketten kurtarma planı hazırlanmış ve test edilmiştir. Amaç, veritabanı bütünlüğünü sağlamak, veri kaybını önlemek ve acil durumlarda hızlı bir şekilde kurtarma işlemleri gerçekleştirebilmektir.

1. Yedekleme Stratejileri

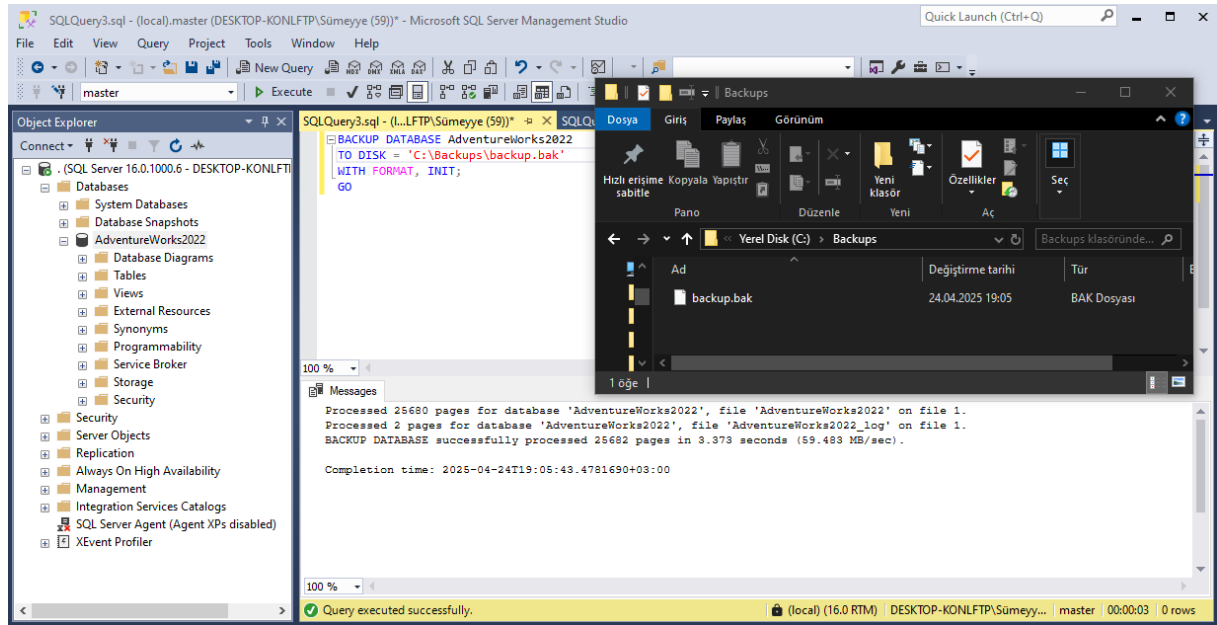
1.1. Tam Yedekleme (Full Backup)

Veritabanının o andaki tam kopyası alındı. BACKUP DATABASE komutu kullanılarak backup.bak dosyası oluşturuldu.

BACKUP DATABASE AdventureWorks2022

TO DISK = 'C:\Backups\AW_full.bak'

WITH FORMAT, INIT;



Şekil 1: backup.bak dosyası başarıyla oluşturulması ve yedek alınması.

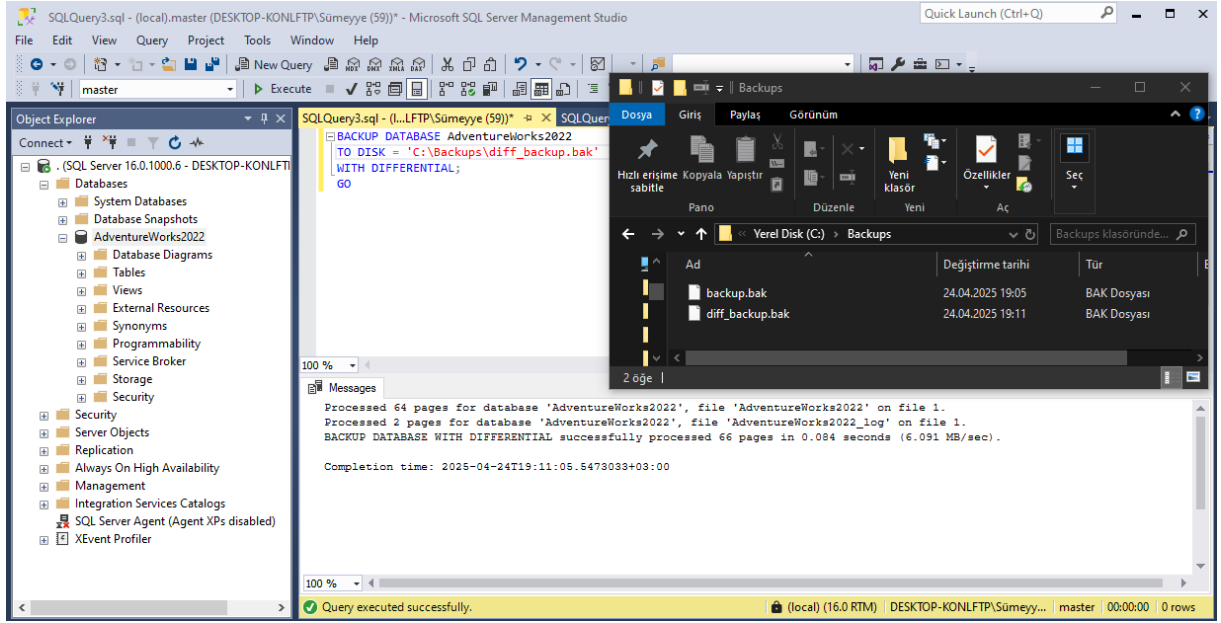
1.2. Artımlı Yedekleme (Differential Backup)

Full yedekten sonra değişen veriler hızlıca yedeklendi. BACKUP DATABASE ... WITH DIFFERENTIAL komutu çalıştırıldı ve diff_backup.bak oluşturuldu.

BACKUP DATABASE AdventureWorks2022

TO DISK = 'C:\Backups\diff_backup.bak'

WITH DIFFERENTIAL;



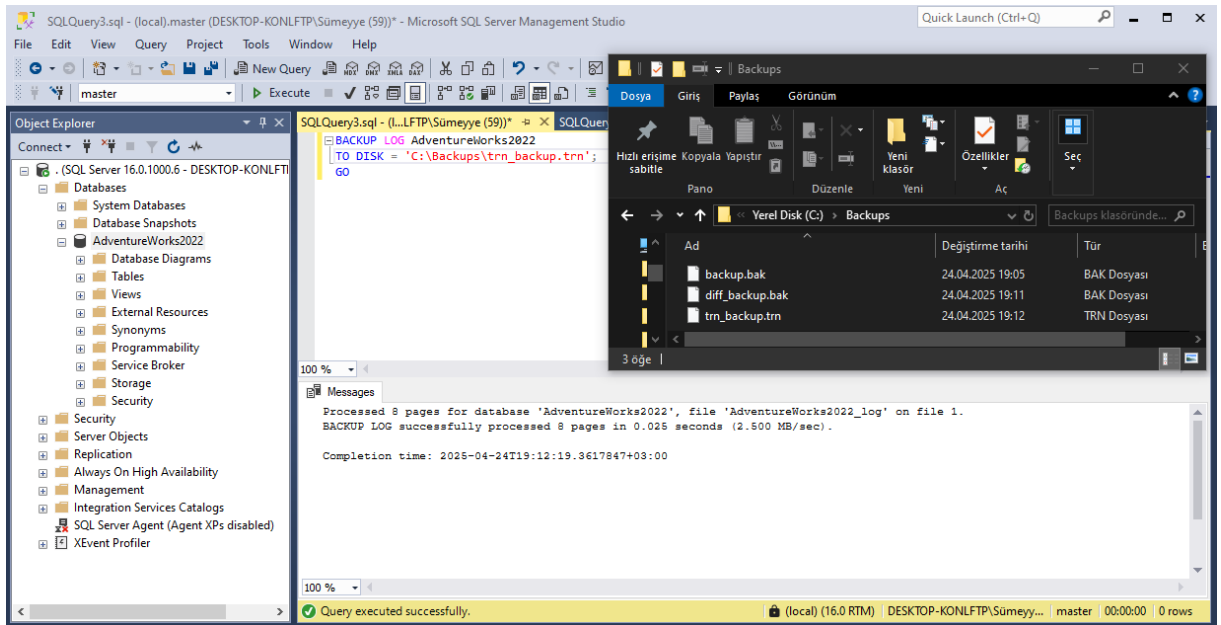
Şekil 2: Değişiklikleri içeren artımlı yedek dosyasının elde edilmesi.

1.3. Transaction Log Yedekleme

Point-in-time kurtarma yeteneği sağlandı. BACKUP LOG komutu ile AW_trn.trn dosyası oluşturuldu.

BACKUP LOG AdventureWorks2022

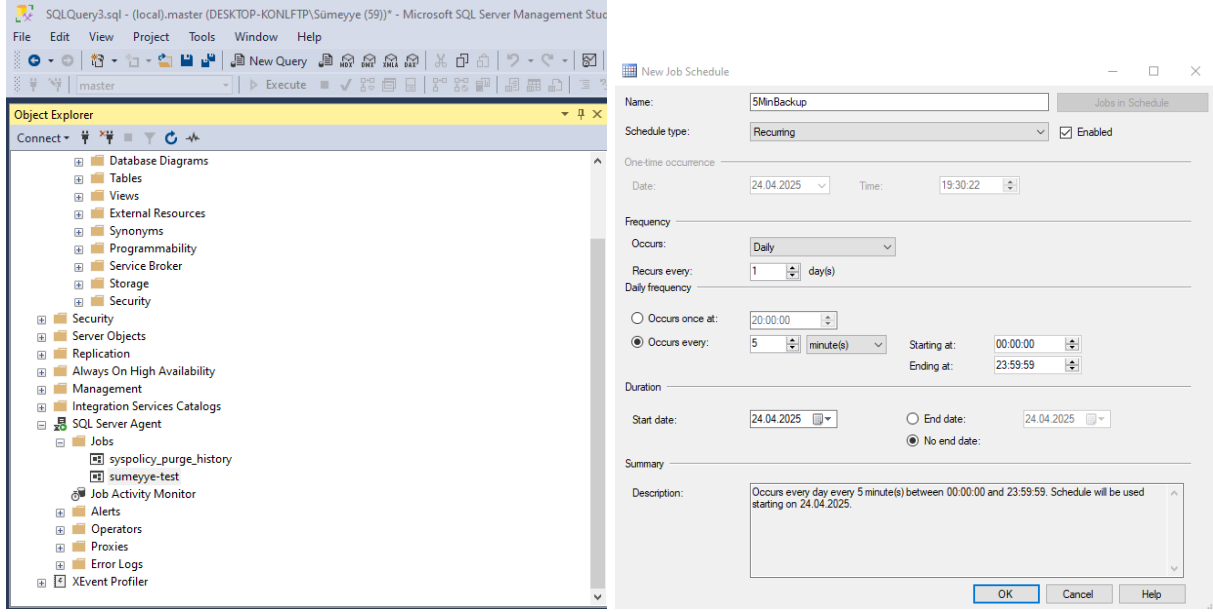
TO DISK = 'C:\Backups\trn_backup.trn';



Şekil 3: Transaction log yedeği başarıyla alınması.

1.4. Yedek Otomasyonu (SQL Server Agent Job)

Yedek işlemleri düzenli ve otomatik olarak yapıldı. sumeyye-test adında bir job oluşturuldu; her 5 dakikada bir full backup alacak şekilde schedule(5MinBackup) tanımlandı.



Şekil 4: Otomatik full yedek job'ın başarıyla oluşturulması.

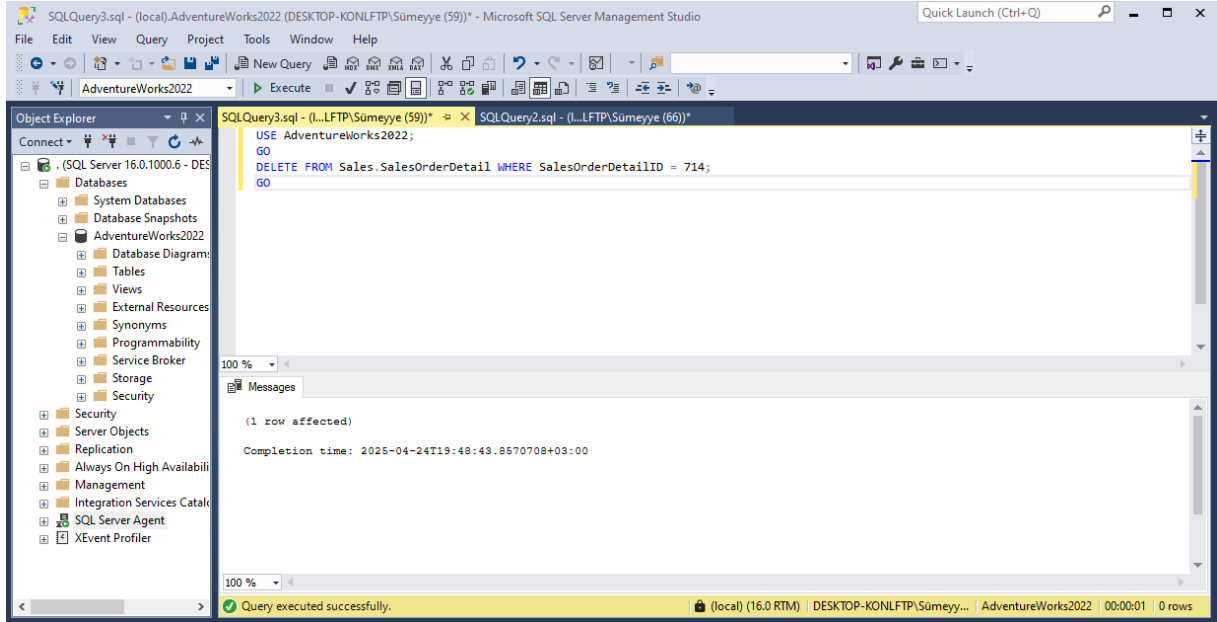
2. Felaketten Kurtarma Senaryoları

2.1. Veri Silme Simülasyonu

Kurtarma adımlarının testi için veri kaybı simüle edildi. SalesOrderDetailID = 714 satırı silindi.

USE AdventureWorks2022;

DELETE FROM Sales.SalesOrderDetail WHERE SalesOrderDetailID = 714;



Şekil 5: Kayıt silinmesi.

2.2. Tam Restore (Full Restore)

Full yedekten veritabanını tamamen geri yüklendi. RESTORE DATABASE komutuyla backup.bak dosyası kullanılarak veritabanı üzerine yazıldı (WITH REPLACE).

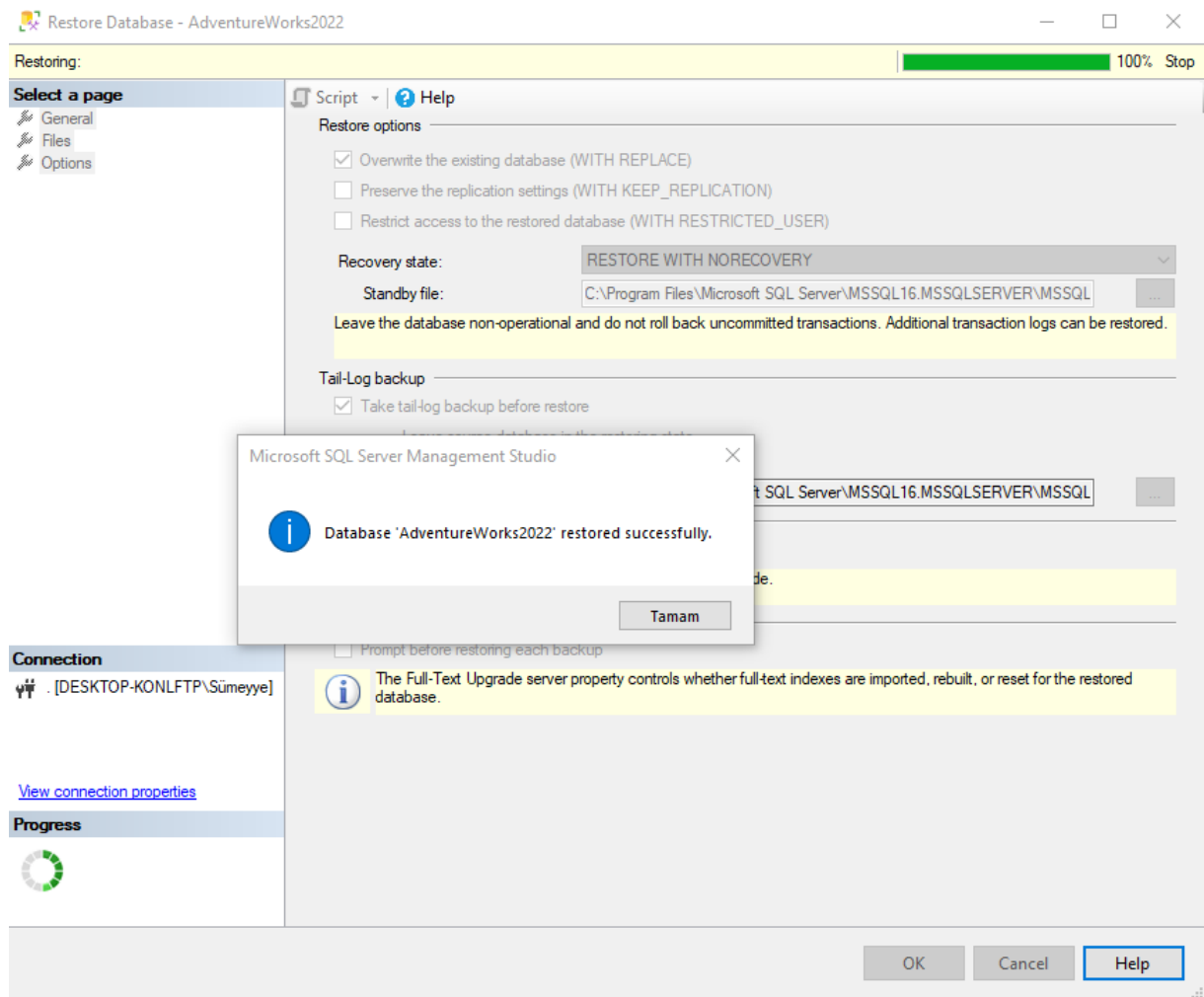
```
ALTER DATABASE AdventureWorks2022 SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
```

```
RESTORE DATABASE AdventureWorks2022
```

```
FROM DISK = 'C:\Backups\backup.bak'
```

```
WITH REPLACE;
```

```
ALTER DATABASE AdventureWorks2022 SET MULTI_USER;
```



Şekil 6: Veritabanının tam olarak geri yüklenmesi

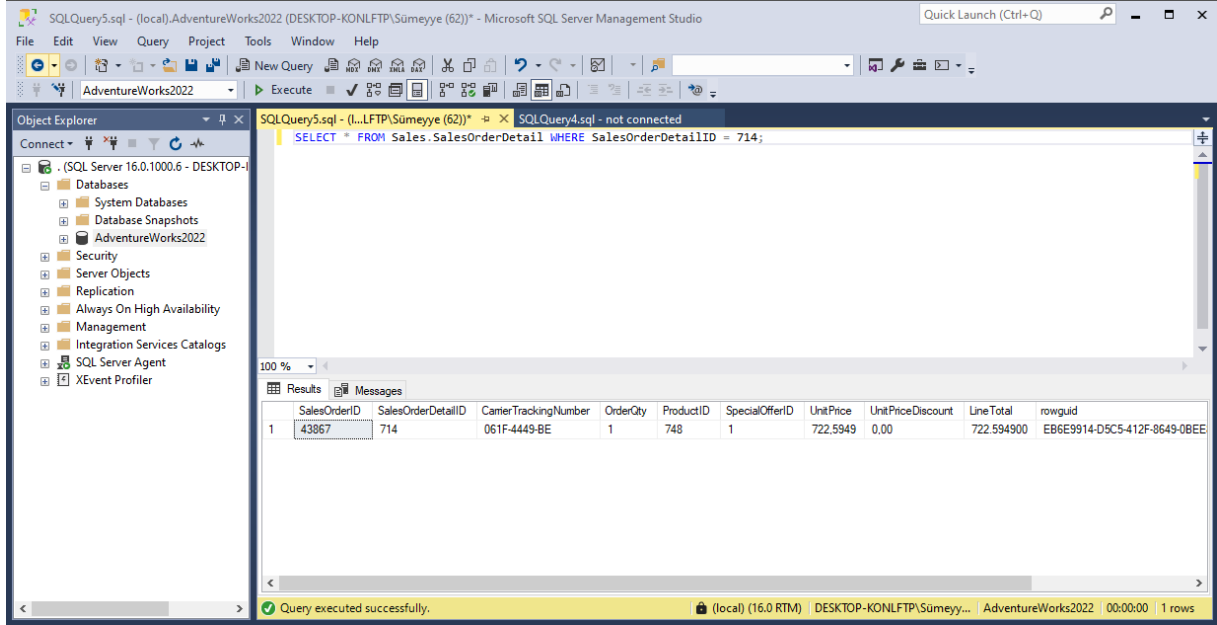
2.3. Zaman Noktası Kurtarma (Point-in-Time Restore)

Belirli bir tarih-saat dilimine dönerek silinen veri kurtarıldı. Transaction log yedeği trn_backup.trn kullanılarak, STOPAT parametresiyle geri yükleme yapıldı ve silinen verinin kurtarıldığı görüldü.

RESTORE LOG AdventureWorks2022

FROM DISK = 'C:\Backups\trn_backup.trn'

WITH STOPAT = '2025-04-24 14:30:00', RECOVERY;



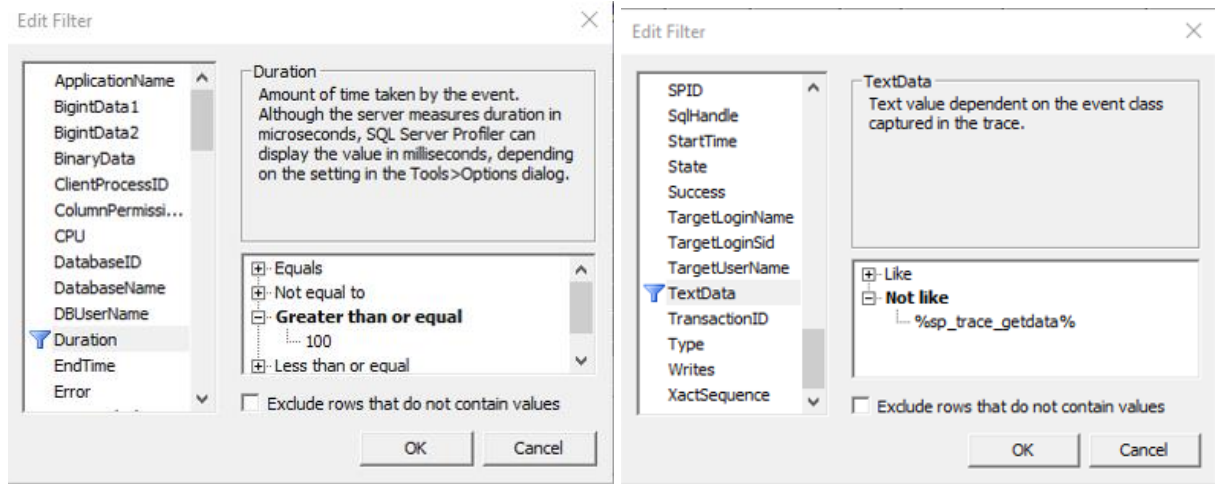
Şekil 7: Söz konusu kayıt geri yüklenerek veri kurtarmanın başarıyla gerçekleştirilmesi.

VERİTABANI PERFORMANS OPTİMİZASYONU VE İZLEME

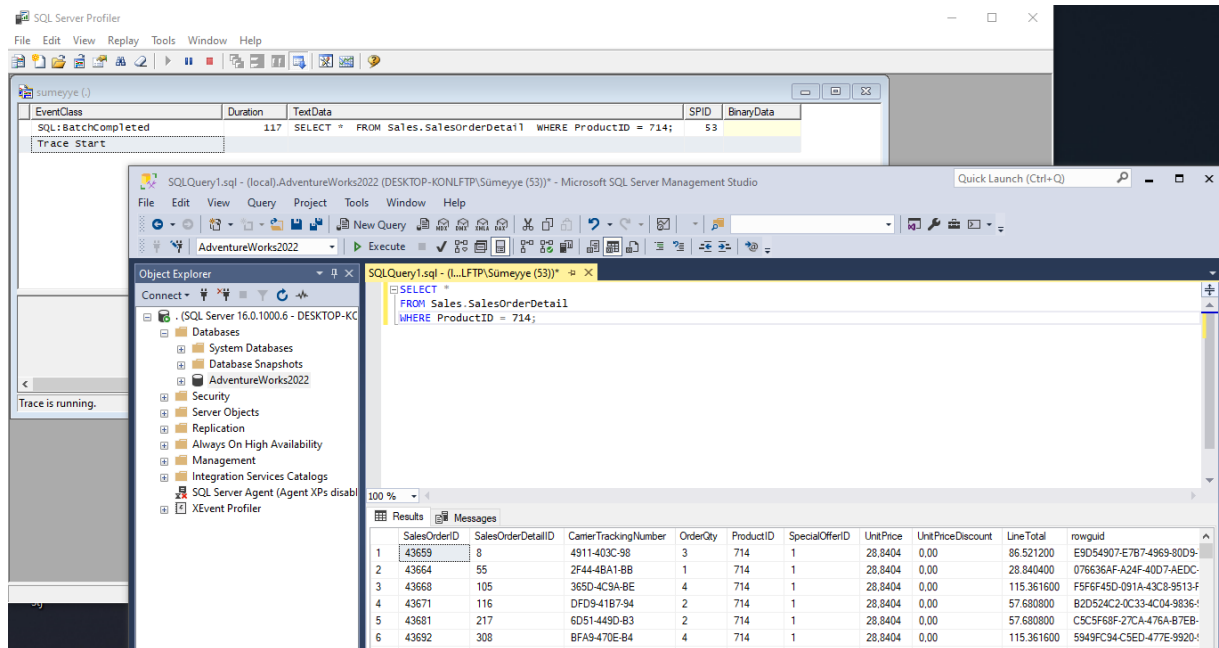
AdventureWorks2022 veritabanındaki yavaş sorgular tespit edilmiş, analiz edilmiş ve indeks optimizasyonu ile performans iyileştirmesi yapılmıştır. Amaç, sorgu sürelerini kısaltmak ve sistem kaynak kullanımını azaltmaktır.

1. SQL Server Profiler ile İzleme

100 milisaniyeden uzun süren sorgular gerçek zamanlı yakalandı. Profiler açıldı, **TSQL_Duration** şablonu seçildi; **Events Selection** → **Column Filters** bölümünde Duration ≥ 100 ms ve TextData NOT LIKE '%sp_trace_getdata%' filtreleri uygulandı ve **Run** ile izleme başlatıldı.



Şekil 8: Events Selection → Column Filters bölümünde Duration ve TextData filtreleri.

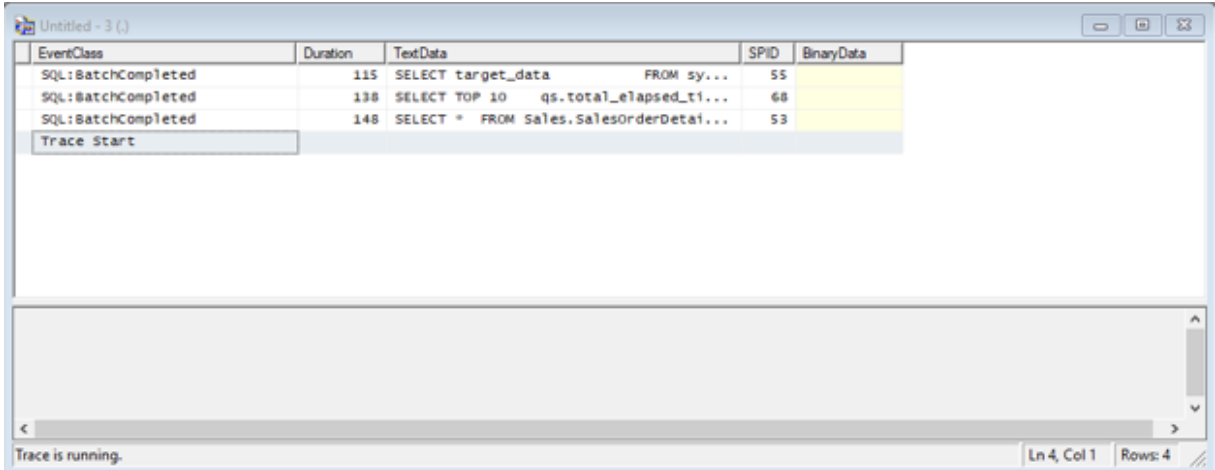


Şekil 9: SSMS üzerinden örnek sorgular gönderilerek slow query'lerin Profiler'da listelenmesi.

2. Dynamic Management Views (DMV) ile Sorgu İstatistikleri

Cache’de tutulan yavaş sorgular sorgulanarak en yüksek ortalama süreye sahip olanı belirlendi. Aşağıdaki script çalıştırıldı:

```
SELECT TOP 10  
    qs.total_elapsed_time/qs.execution_count AS avg_duration_ms,  
    qs.execution_count,  
    qt.text AS query_text  
FROM sys.dm_exec_query_stats qs  
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) qt  
WHERE qt.text NOT LIKE 'sys.%'  
ORDER BY avg_duration_ms DESC;
```



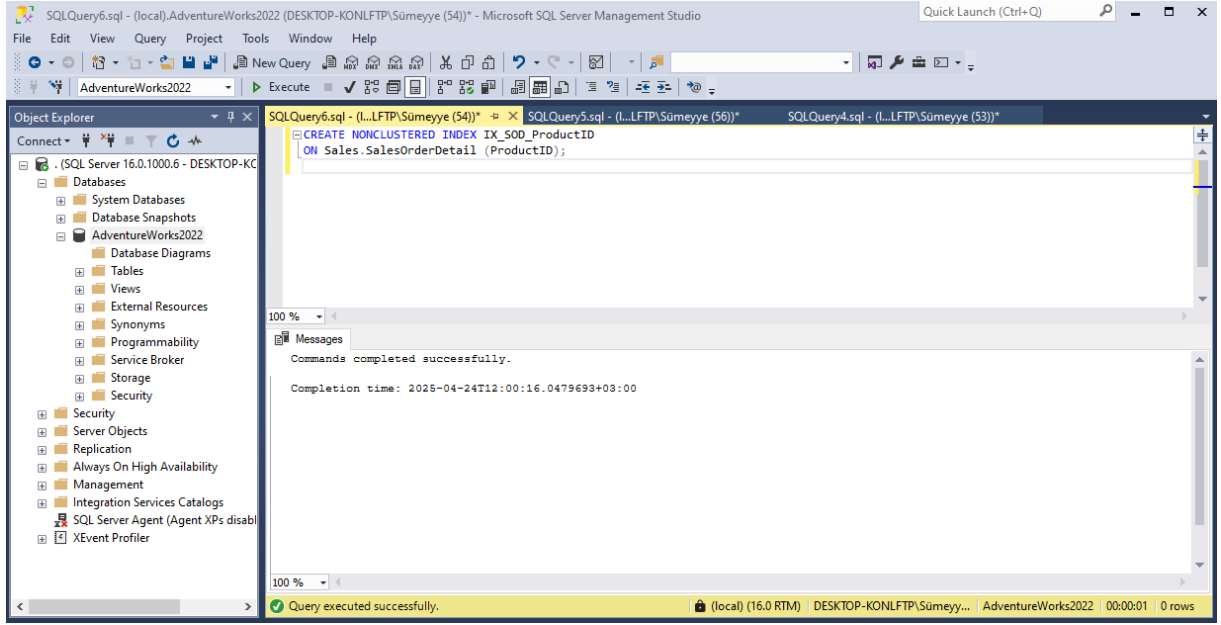
EventClass	Duration	TextData	SPID	BinaryData
SQL: BatchCompleted	115	SELECT target_data FROM sy...	55	
SQL: BatchCompleted	138	SELECT TOP 10 qs.total_elapsed_ti...	68	
SQL: BatchCompleted	148	SELECT * FROM Sales.SalesOrderDetail...	53	
Trace Start				

Şekil 10: En yavaş sorgunun SELECT * FROM Sales.SalesOrderDetail WHERE ProductID = 714; olarak saptanması.

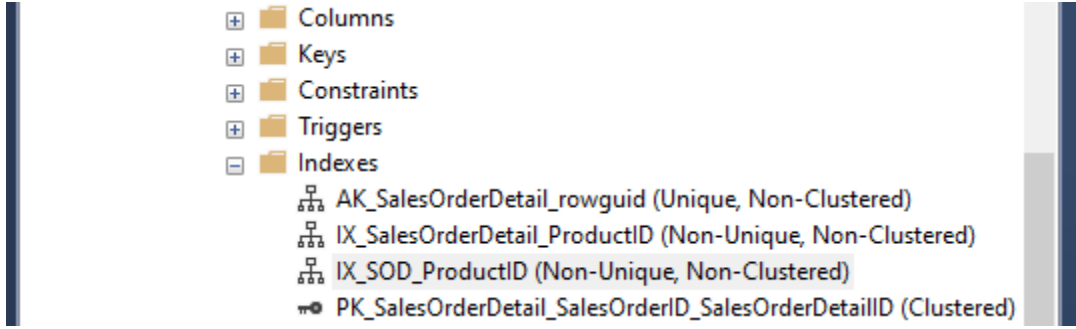
3. İndeks Oluşturma

Belirlenen sorgunun çalıştırma maliyeti düşürüldü. Aşağıdaki komut çalıştırıldı:

```
CREATE NONCLUSTERED INDEX IX_SOD_ProductID  
ON Sales.SalesOrderDetail (ProductID);
```

Şekil 11: T-SQL ile indeks oluşturma script'i ve başarı mesajı.

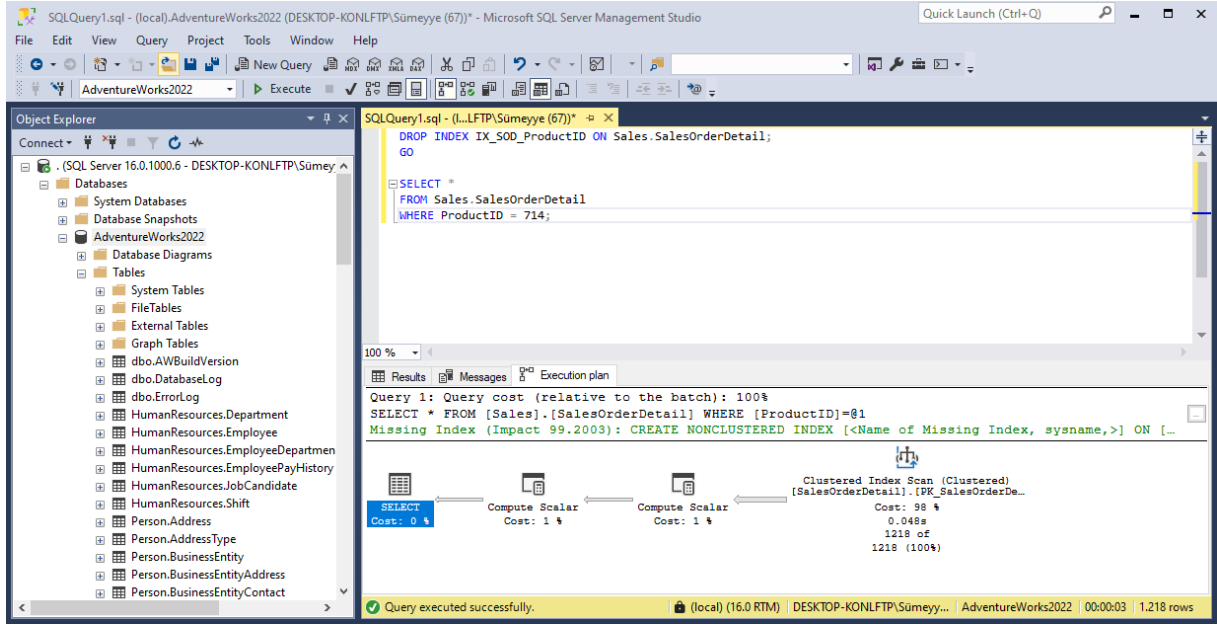


Şekil 12: IX_SOD_ProductID indeksinin başarıyla eklenmesi.

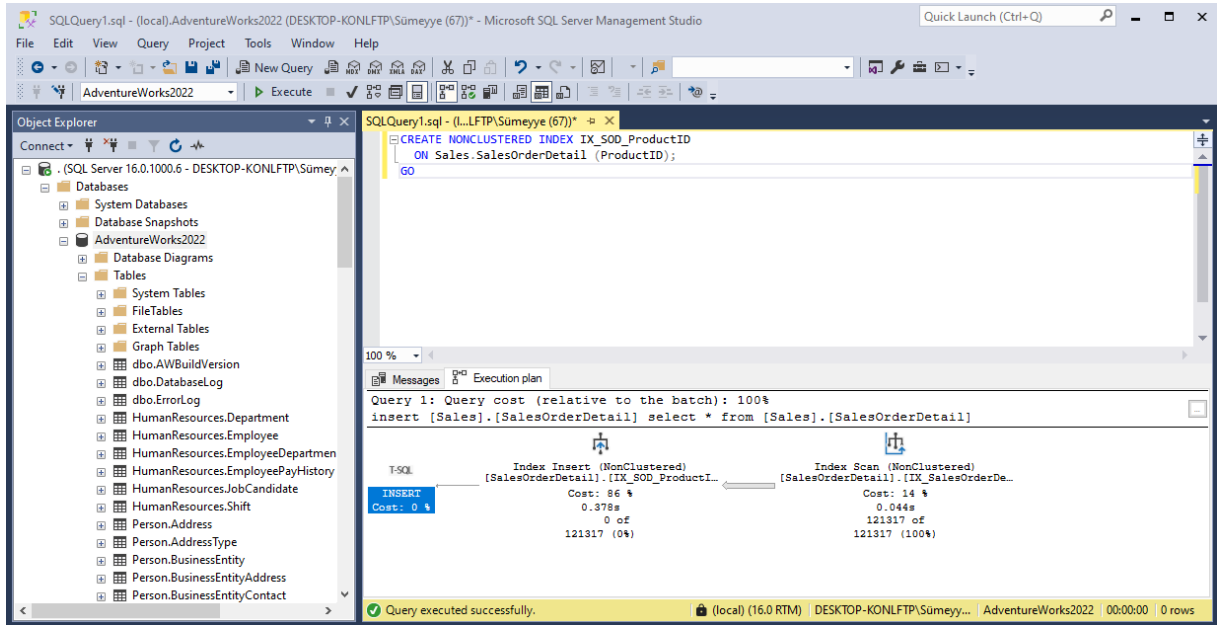
4. Execution Plan Testi

İndeksin sorgu planına etkisi görsel ve metrik olarak doğrulandı. **Include Actual Execution Plan** aktif edilerek sorgu önce ve sonra çalıştırıldı.

- Önce: **SELECT * FROM Sales.SalesOrderDetail WHERE ProductID = 714;** → **Index Scan**
- Sonra: Aynı sorgu → **Index Seek**



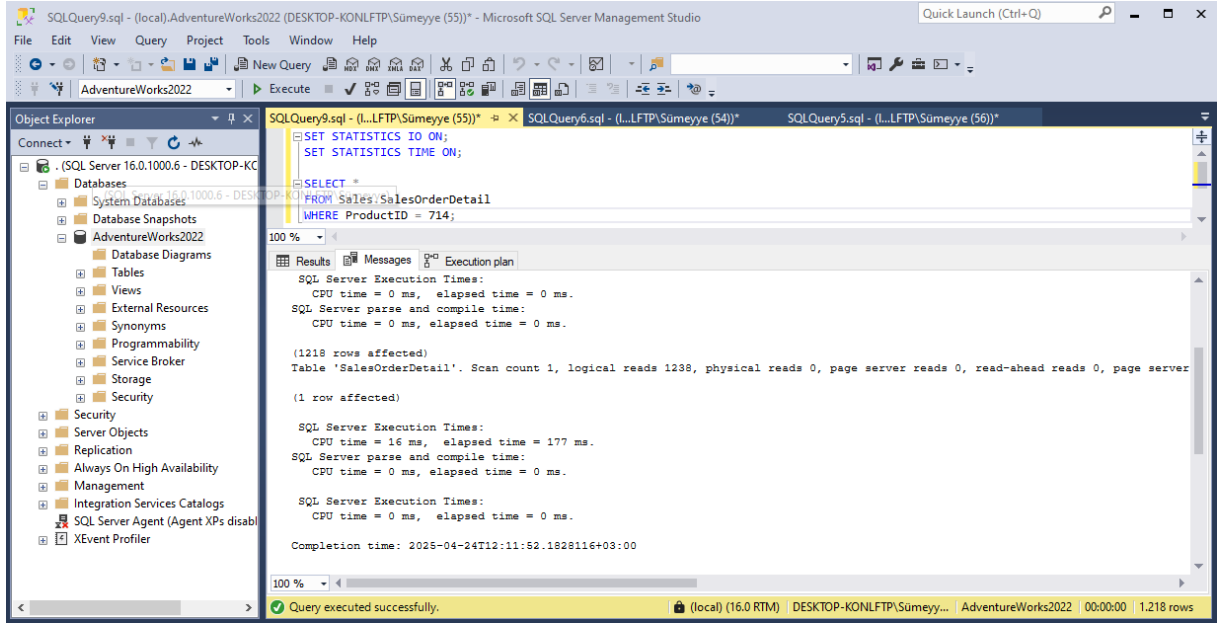
Şekil 13: İndeks öncesi Execution Plan – Index Scan.



Şekil 14: İndeks sonrası Execution Plan – Index Seek.

4. STATISTICS TIME & IO

Disk okuma ve işlem sürelerindeki iyileşme sayısal olarak belgelendi. SET STATISTICS IO ON; SET STATISTICS TIME ON; komutlarıyla sorgu çalıştırıldı.



Şekil 15: STATISTICS IO/TIME sonuçları.

VERİTABANI GÜVENLİĞİ VE ERİŞİM KONTROLÜ

AdventureWorks2022 veritabanı üzerinde güvenlik politikasının uygulanması, kullanıcı erişimlerinin kontrolü, kolon bazlı şifreleme ve etkinlik denetimi (audit) adımları yapılmıştır. Amaç, yetkisiz erişimi engellemek, hassas verinin korunmasını sağlamak ve kullanıcı etkinliklerini izlemektir.

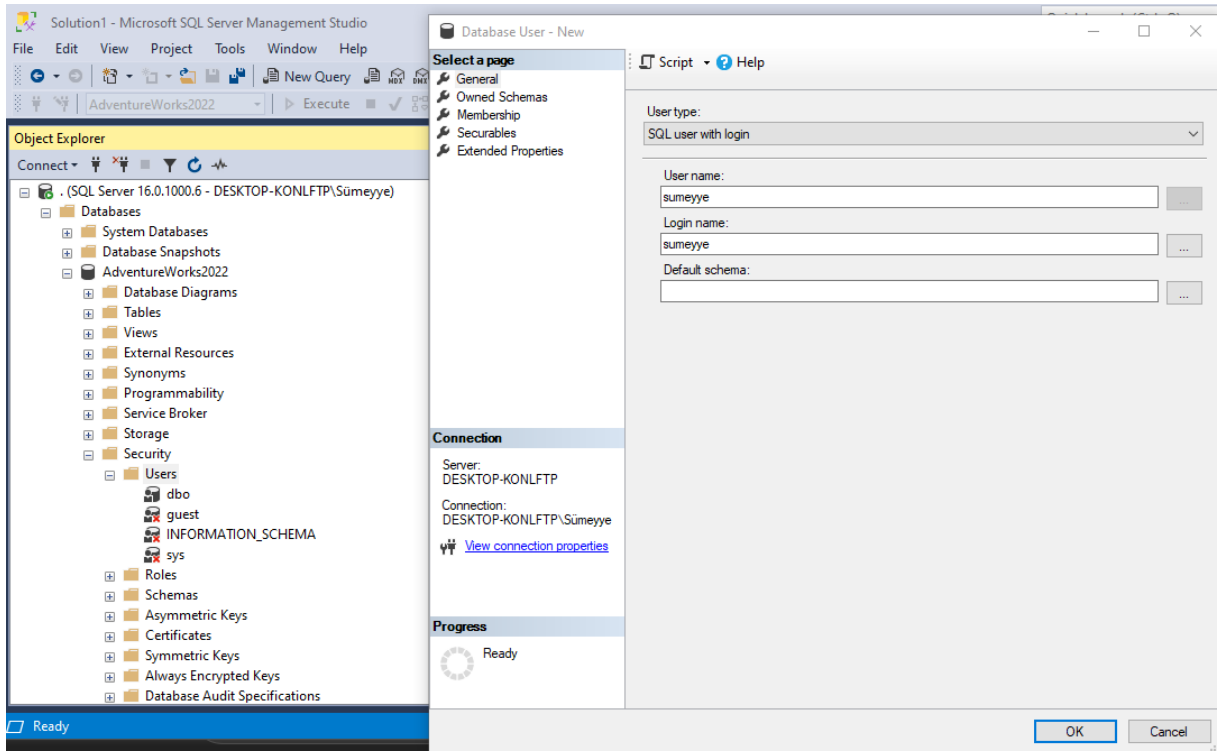
1. Login ve Database User Oluşturma

Uygulama kullanıcıları için ayrı SQL login ve veritabanı kullanıcı hesabı tanımlandı..

- Security → Logins → New Login... sihirbazında **sumeyye** SQL Server login'i oluşturuldu, parola atandı.
- AdventureWorks2022 → Security → Users → New User... sihirbazı ile **sumeyye** veritabanı kullanıcısı eklendi.

The screenshot shows the 'Login - New' dialog box in SQL Server Enterprise Manager. The 'General' tab is selected. The 'Login name' is 'sumeyye'. The 'Authentication' section has 'SQL Server authentication' selected. The 'Password' and 'Confirm password' fields are filled with dots. The 'Enforce password policy', 'Enforce password expiration', and 'User must change password at next login' checkboxes are checked. The 'Mapped to certificate', 'Mapped to asymmetric key', and 'Map to Credential' options are not selected. The 'Default database' is 'AdventureWorks2022' and the 'Default language' is '<default>'. The 'Progress' section shows 'Ready'.

Şekil 16: New Login penceresi.

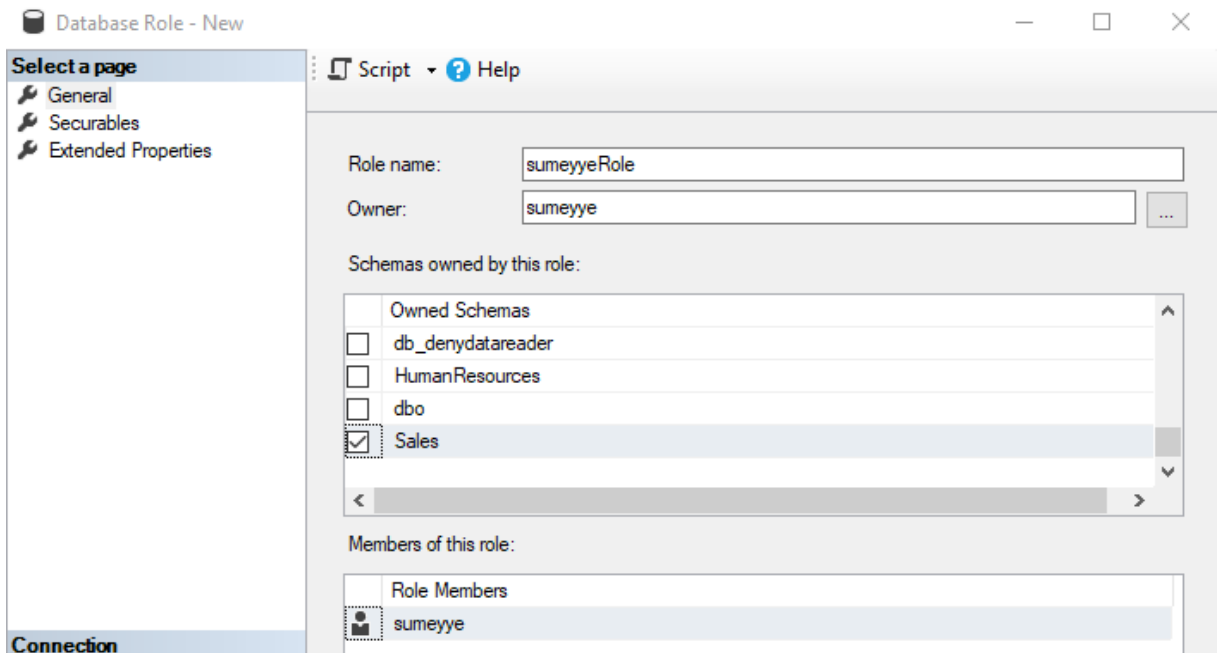


Şekil 17: New User penceresi.

2. Rol Tabanlı Erişim Kontrolü

Belirli tablolar için sadece SELECT yetkisine sahip bir rol oluşturmak.

- AdventureWorks2022 → Security → Roles → Database Roles → New Database Role... ile **sumeyyeRole** rolü oluşturuldu.
- Members sekmesinden sumeyye rolün üyesi yapıldı.

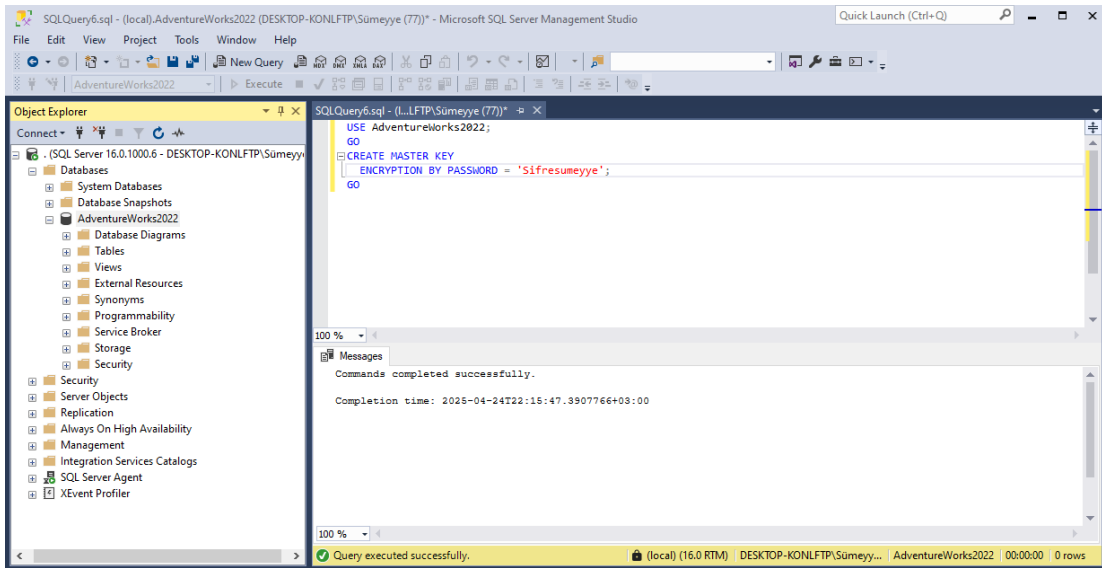


Şekil 18: New Database Role penceresi.

3. Şifreleme (T-SQL Symmetric Key)

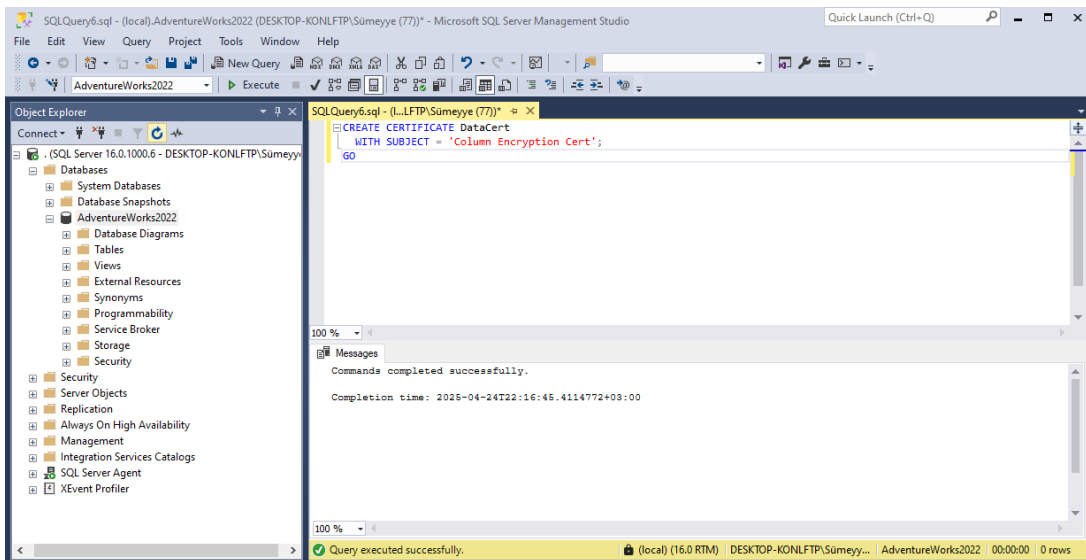
Hassas LineTotal sütunu şifrelendi ve sadece yetkili anahtar açıldığında okunabilir hale getirildi.

- Master Key oluşturuldu:
USE AdventureWorks2022;
GO
CREATE MASTER KEY
ENCRIPTION BY PASSWORD = 'Sifresumeyye';
GO



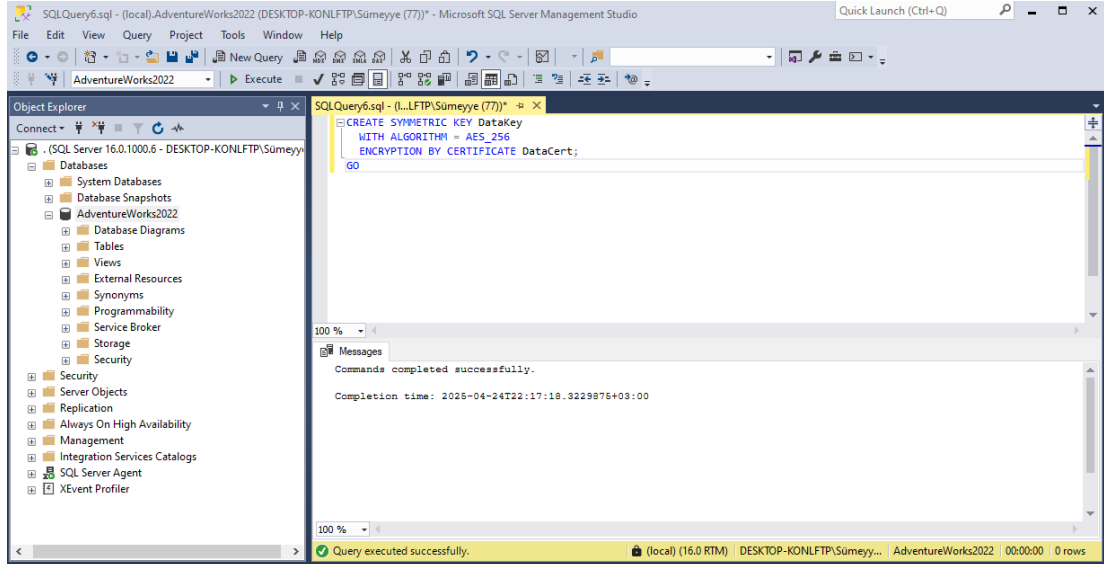
Şekil 19: Master key oluşturulması.

- Sertifika oluşturuldu:
CREATE CERTIFICATE DataCert
WITH SUBJECT = 'Column Encryption Cert';
GO



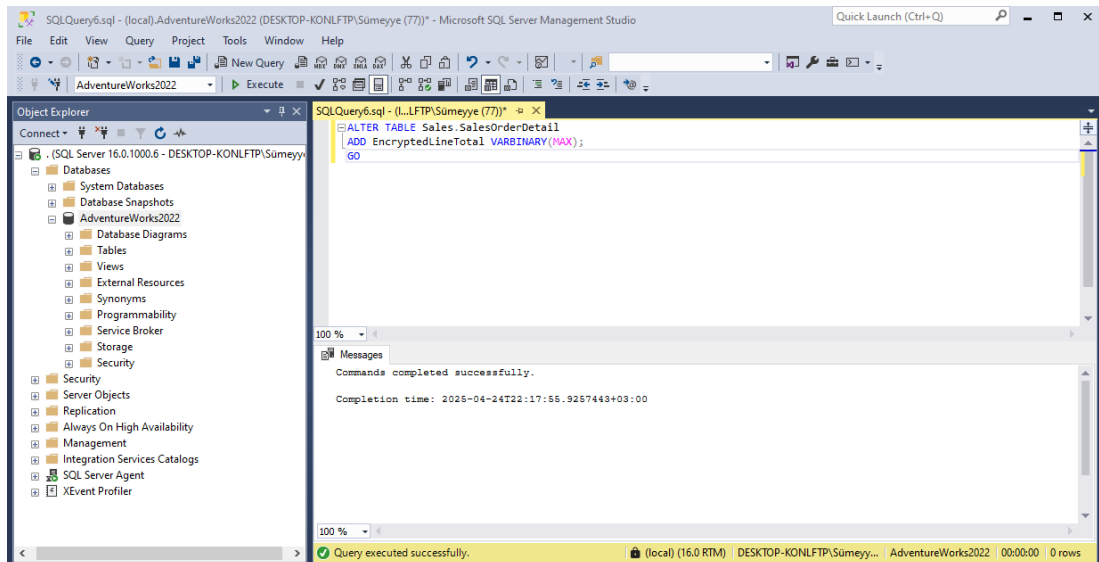
Şekil 20: Sertifika oluşturulması.

- Symmetric Key oluşturuldu:
CREATE SYMMETRIC KEY DataKey
WITH ALGORITHM = AES_256
ENCRYPTION BY CERTIFICATE DataCert;
GO



Şekil 21: Symmetric key oluşturulması.

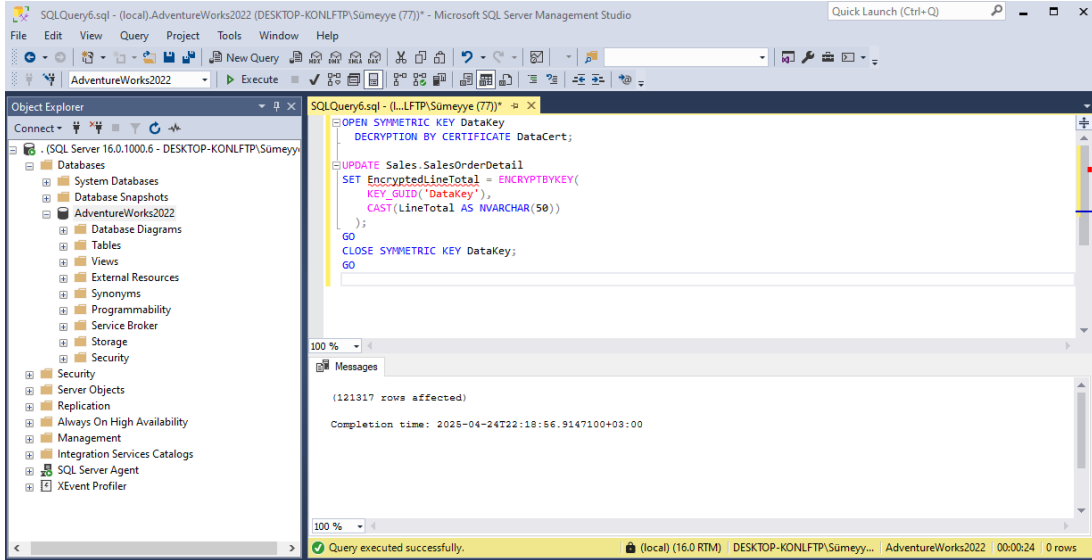
- Yeni kolon eklendi:
ALTER TABLE Sales.SalesOrderDetail
ADD EncryptedLineTotal VARBINARY(MAX);
GO



Şekil 22: Yeni kolon eklenmesi.

- Veri şifreleme işlemi gerçekleştirildi:
OPEN SYMMETRIC KEY DataKey
DECRYPTION BY CERTIFICATE DataCert;
UPDATE Sales.SalesOrderDetail

```
SET EncryptedLineTotal = ENCRYPTBYKEY(  
    KEY_GUID('DataKey'),  
    CAST(LineTotal AS NVARCHAR(50))  
);  
CLOSE SYMMETRIC KEY DataKey;  
GO
```

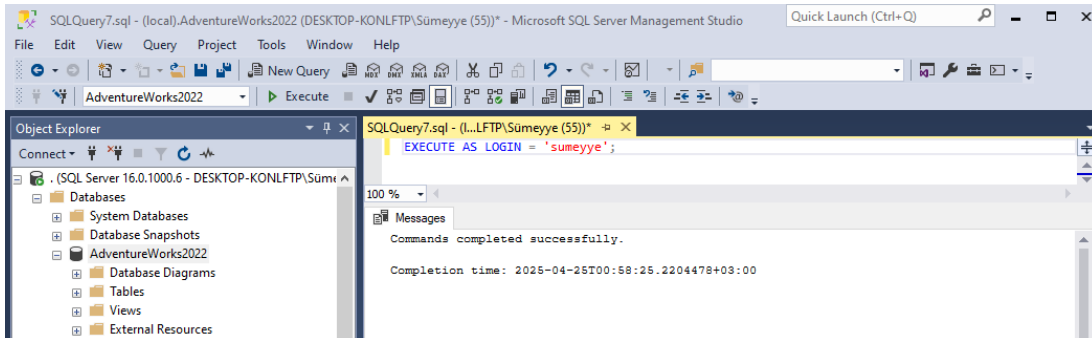


Şekil 23: Veri şifreleme.

4. SQL Injection Testi

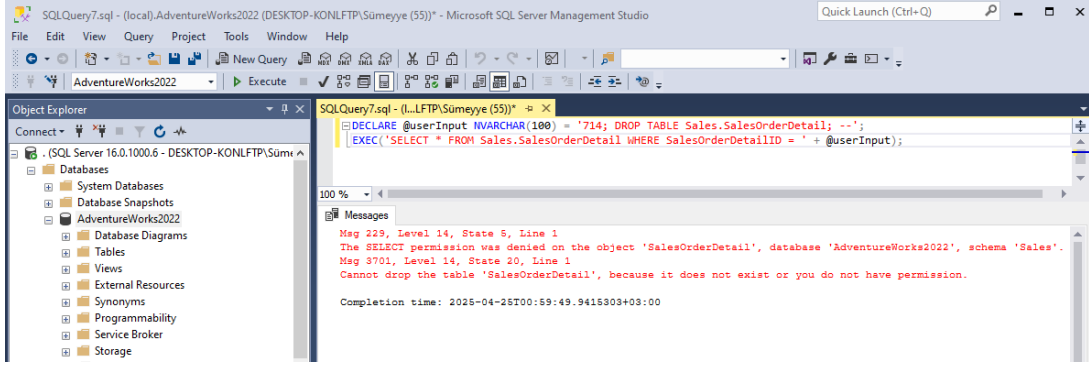
Kullanıcı hakkı kısıtlamasının doğru çalıştığı doğrulandı..

- **EXECUTE AS LOGIN = 'sumeyye';** ile sumeyye kimliğine geçildi.



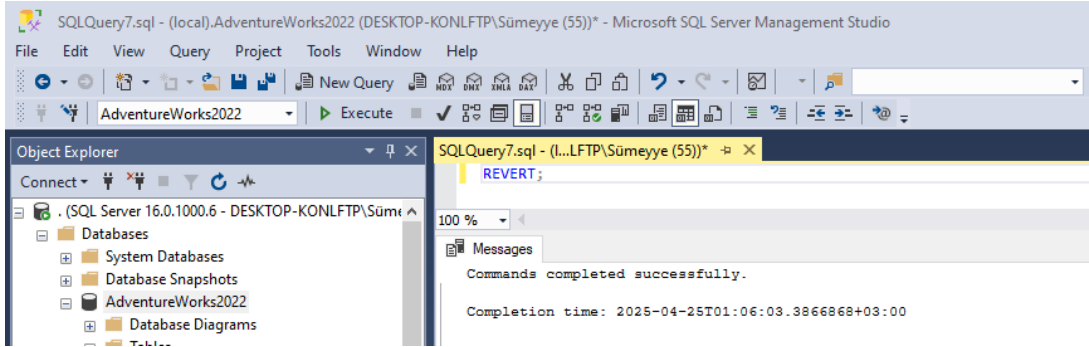
Şekil 24: sumeyye kimliğine geçilmesi.

- Zararlı input simülasyonu:
DECLARE @userInput NVARCHAR(100) = '714; DROP TABLE Sales.SalesOrderDetail; --';
EXEC(SELECT * FROM Sales.SalesOrderDetail WHERE SalesOrderDetailID = ' + @userInput);



Şekil 25: SQL Injection denemesinin reddedilmesi

- Tablonun silinmesi revert komutu ile engellendi.

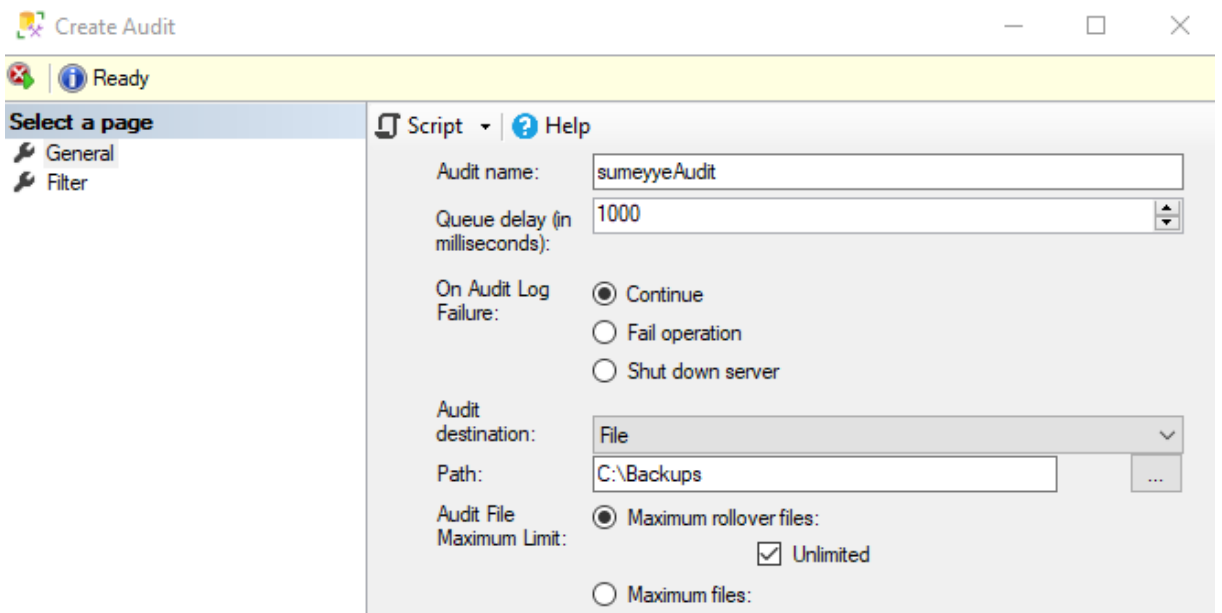


Şekil 26: **REVERT**; komutu ile tablonun silinmesinin engellenmesi.

5. SQL Server Audit ile Activity Loglama

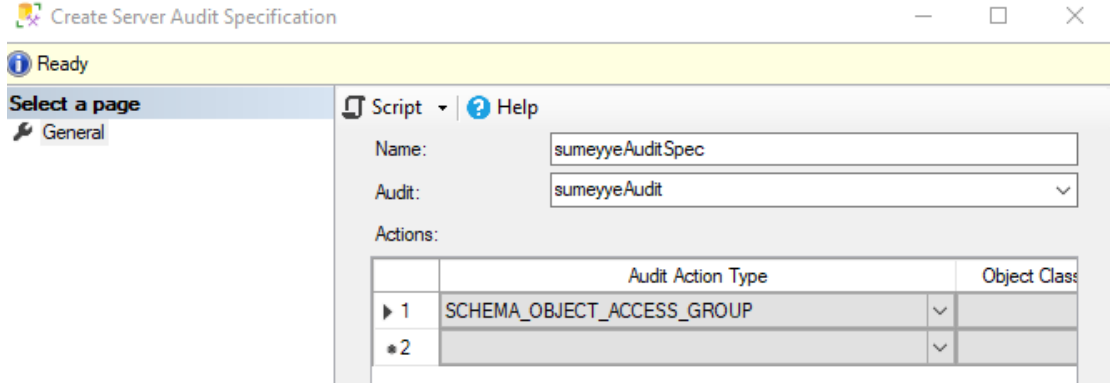
Kullanıcı SELECT işlemleri ve hatalı erişim denemeleri denetlenerek loglandı.

- Server Audit oluşturuldu:
Security → Audits → New Audit... ile **sumeyyeAudit** tanımlandı. **Audit destination:** File seçildi, **File Path:** C:\Backups olarak belirtildi.



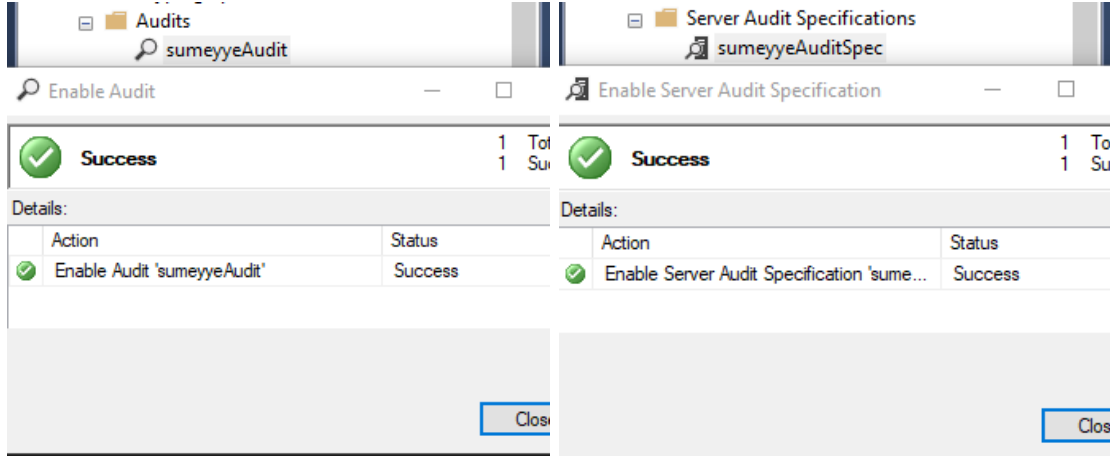
Şekil 27: sumeyyeAudit tanımlanması.

- Server Audit Specification oluşturuldu:
Security → Server Audit Specifications → New Server Audit Specification... ile **SelectAuditSpec** tanımlandı. **Audit:** sumeyyeAuditSpec seçildi. **Action Type:** SCHEMA_OBJECT_ACCESS_GROUP seçildi. **Object Class** alanı boş bırakıldı.



Şekil 28: Server Audit Specification tanımlanması.

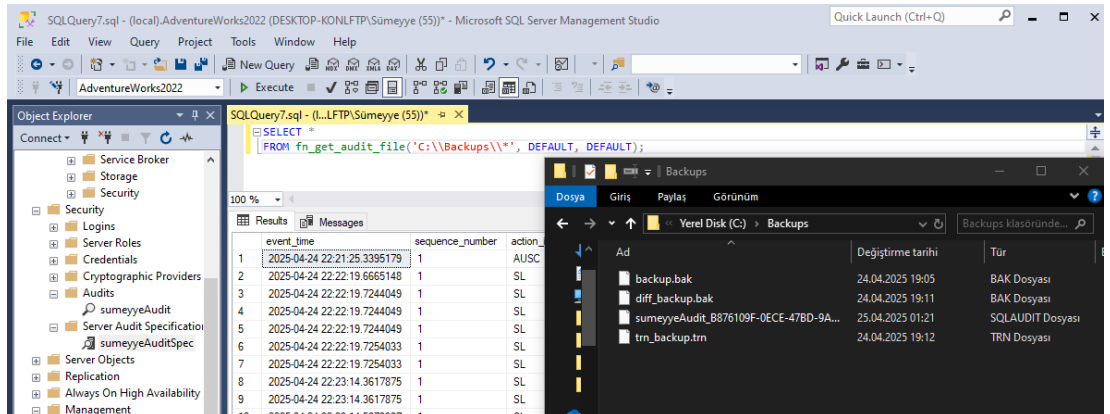
- sumeyyeAudit ve sumeyyeAuditSpec sağ tık → Enable ile etkinleştirildi.



Şekil 29: sumeyyeAudit ve sumeyyeAuditSpec aktifleştirilmesi.

- Audit log kayıtları aşağıdaki komutla görüntülendi:

SELECT *
FROM fn_get_audit_file('C:\\Backups\\', DEFAULT, DEFAULT);



Şekil 30: Audit log kayıtlarının görüntülenmesi.