**ANKARA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

BLM 4522 AĞ TABANLI PARALEL DAĞITIM SİSTEMLERİ

PROJE RAPORU

FATİH LİMANLIK

20291183

GitHub: fatihlimanlik

Proje işlemlerini **AdventureWork2022** veritabanı üzerinde gerçekleştirdim.

**1.VERİTABANI PERFORMANS OPTİMIZASYONU VE İZLEME**

**a. Veritabanı İzleme:**

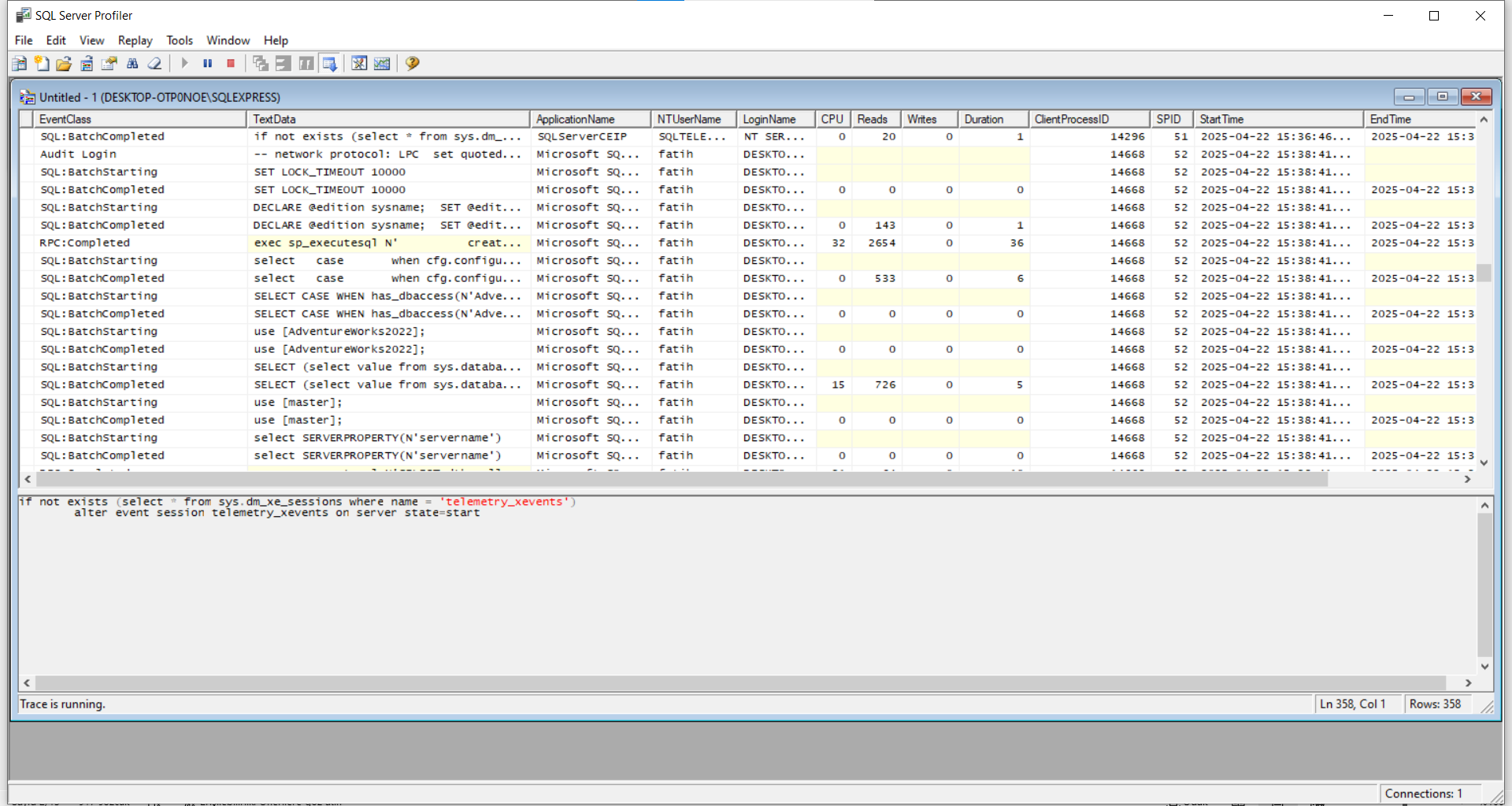
SQL Server Profiler Kullanımı;

1. SQL Server Management Studio (SSMS) → Tools> SQL Server Profiler açılır.
2. Yeni bir trace başlatılır:

"TSQL\_Duration" gibi bir şablon seçilir.

Filtre olarak Duration> 1000ms gibi uzun süren sorgular filtrelenir.

1. İzlemeye başlayarak hangi sorguların yavaş çalıştığını analiz edilir.

****

DMV ile Canlı Sorgu Performansı Takibi için;

en çok CPU kullanan 10 sorgu için aşağıdaki sorgu çalıştırılır.

*SELECT TOP 10*

*total\_worker\_time/1000 AS TotalCPU\_ms,*

*execution\_count,*

*total\_elapsed\_time/1000 AS TotalElapsedTime\_ms,*

*SUBSTRING(qt.text, (qs.statement\_start\_offset/2)+1,*

*((CASE qs.statement\_end\_offset*

*WHEN -1 THEN DATALENGTH(qt.text)*

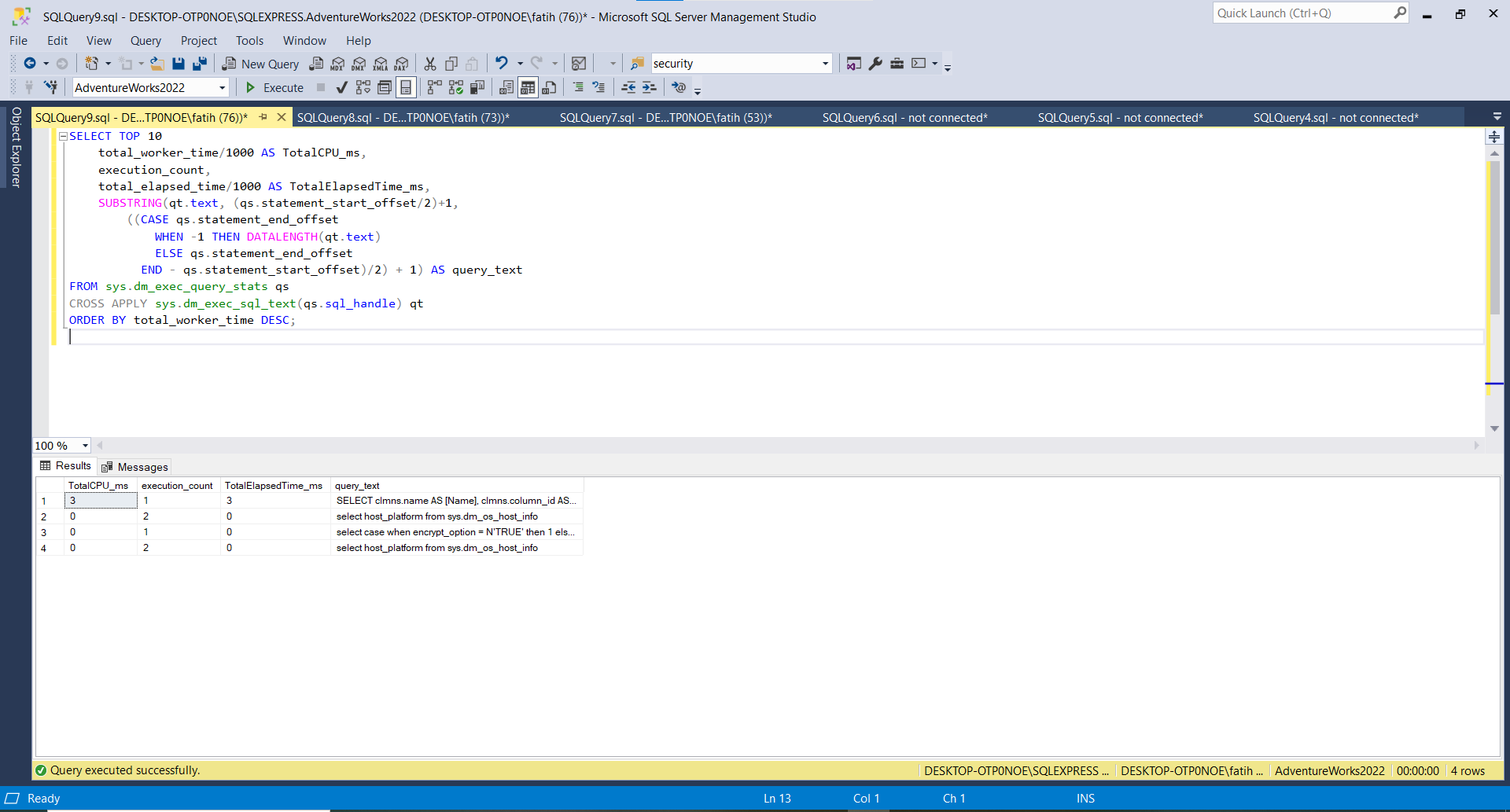
*ELSE qs.statement\_end\_offset*

*END - qs.statement\_start\_offset)/2) + 1) AS query\_text*

*FROM sys.dm\_exec\_query\_stats qs*

*CROSS APPLY sys.dm\_exec\_sql\_text(qs.sql\_handle) qt*

*ORDER BY total\_worker\_time DESC;*

**

DVM ile Disk kullanımı yüksek tabloları görüntülemek için aşağıdaki sorgu çalıştırılır.

*SELECT*

*t.NAME AS TableName,*

*s.name AS SchemaName,*

*p.rows AS RowCounts,*

*SUM(a.total\_pages) \* 8 AS TotalSpaceKB,*

*SUM(a.used\_pages) \* 8 AS UsedSpaceKB*

*FROM sys.tables t*

*INNER JOIN sys.indexes i ON t.OBJECT\_ID = i.object\_id*

*INNER JOIN sys.partitions p ON i.object\_id = p.OBJECT\_ID AND i.index\_id = p.index\_id*

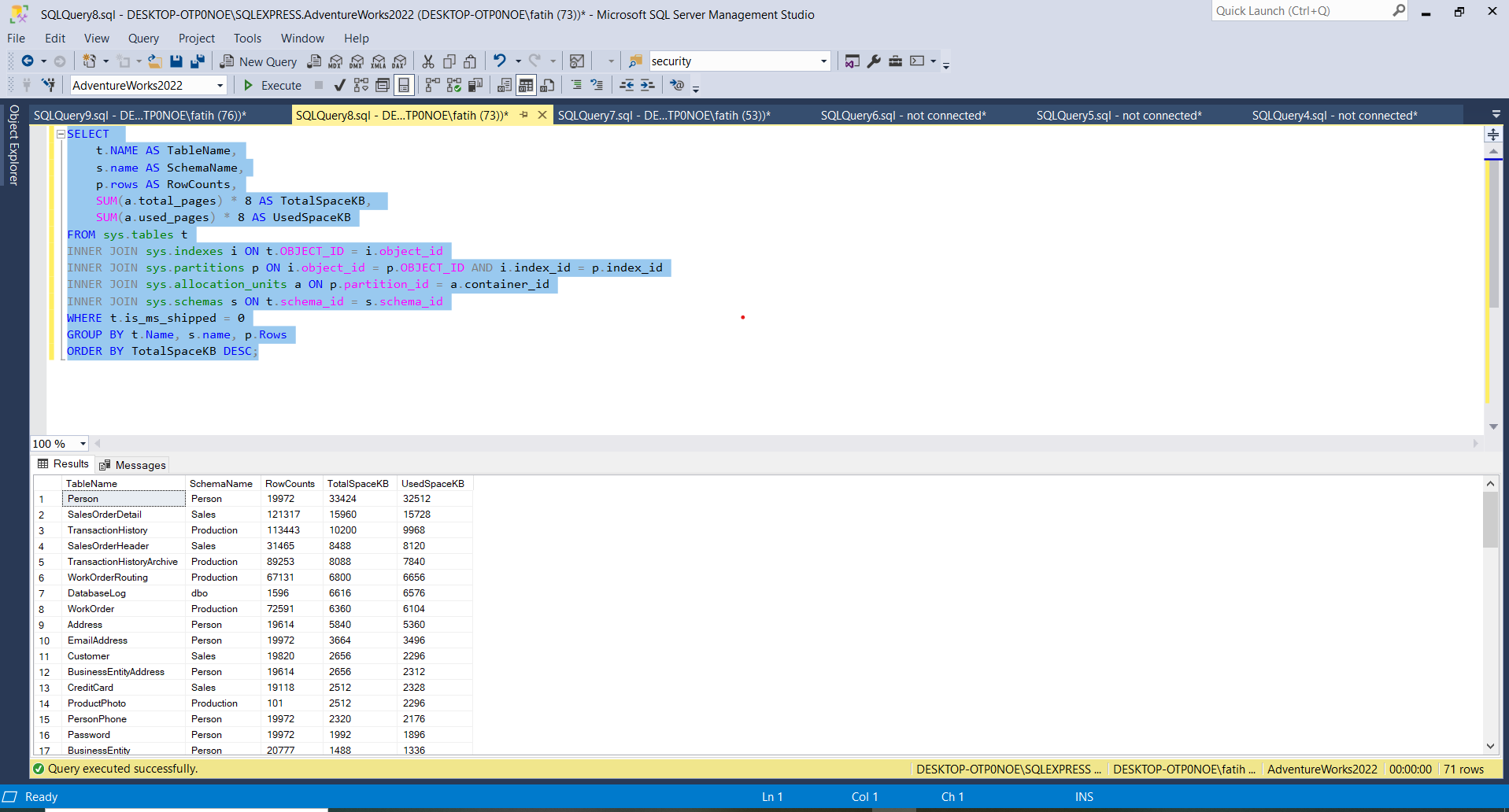
*INNER JOIN sys.allocation\_units a ON p.partition\_id = a.container\_id*

*INNER JOIN sys.schemas s ON t.schema\_id = s.schema\_id*

*WHERE t.is\_ms\_shipped = 0*

*GROUP BY t.Name, s.name, p.Rows*

*ORDER BY TotalSpaceKB DESC;*

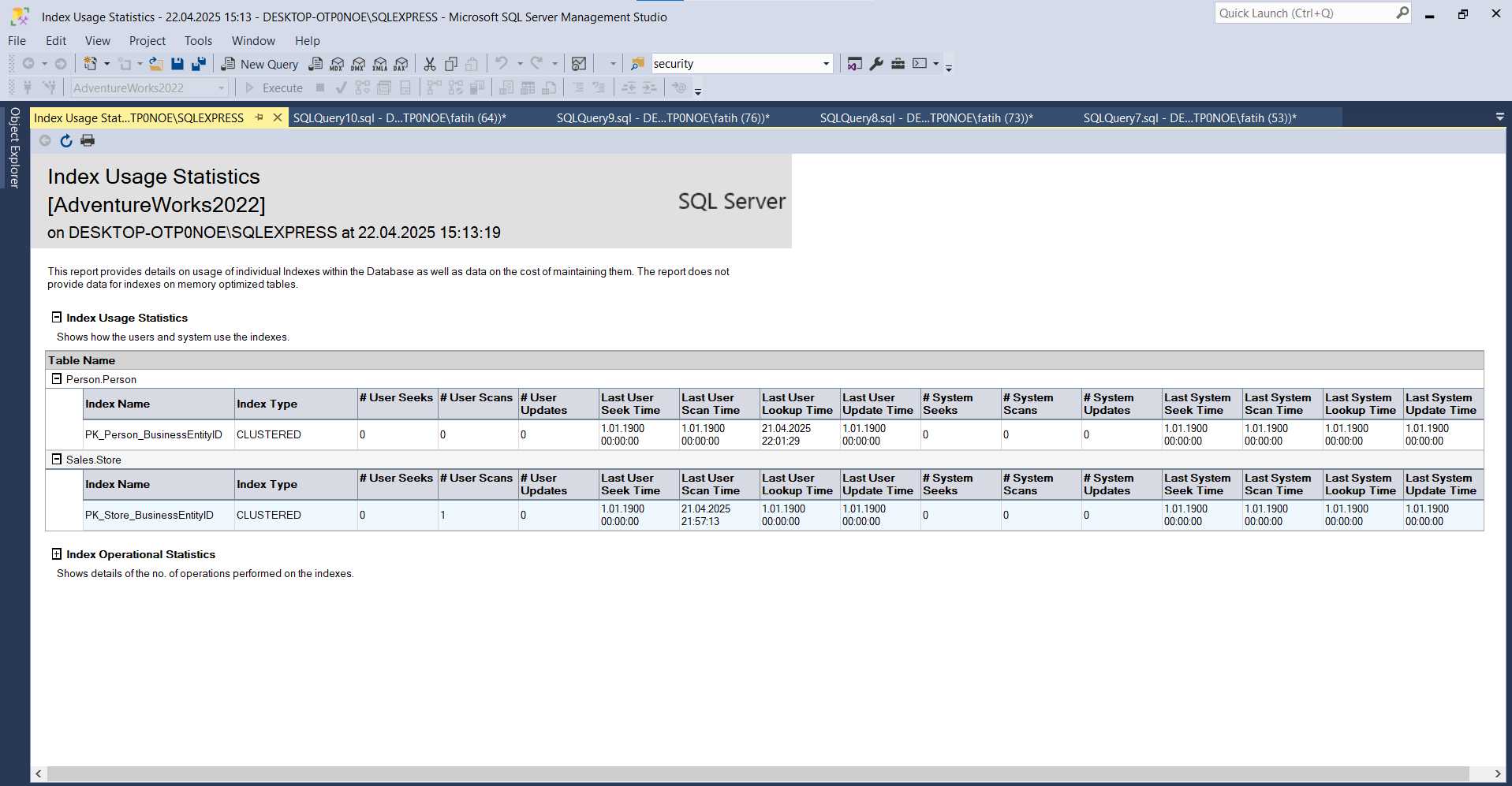
**

**b. İndeks Yönetimi**

İndeks Eksikliklerinin Tespiti için;

1. SSMS'de AdventureWorks2022 veritabanına sağ tıklanır.
2. Reports> Standard Reports> Index Usage Statistics seçilir.

Bu rapor, indekslerin ne kadar kullanıldığı ve eksik indekslerin neler olabileceği hakkında bilgi verir.

****

Aşağıdaki sorgu ile gereksiz veya hiç kullanılmayan indeksler tespit edilir.

*SELECT*

*OBJECT\_NAME(i.object\_id) AS TableName,*

*i.name AS IndexName,*

*i.type\_desc AS IndexType,*

*us.user\_seeks,*

*us.user\_scans,*

*us.user\_lookups,*

*us.user\_updates*

*FROM sys.indexes AS i*

*INNER JOIN sys.objects AS o ON i.object\_id = o.object\_id*

*LEFT JOIN sys.dm\_db\_index\_usage\_stats AS us ON i.object\_id = us.object\_id AND i.index\_id = us.index\_id*

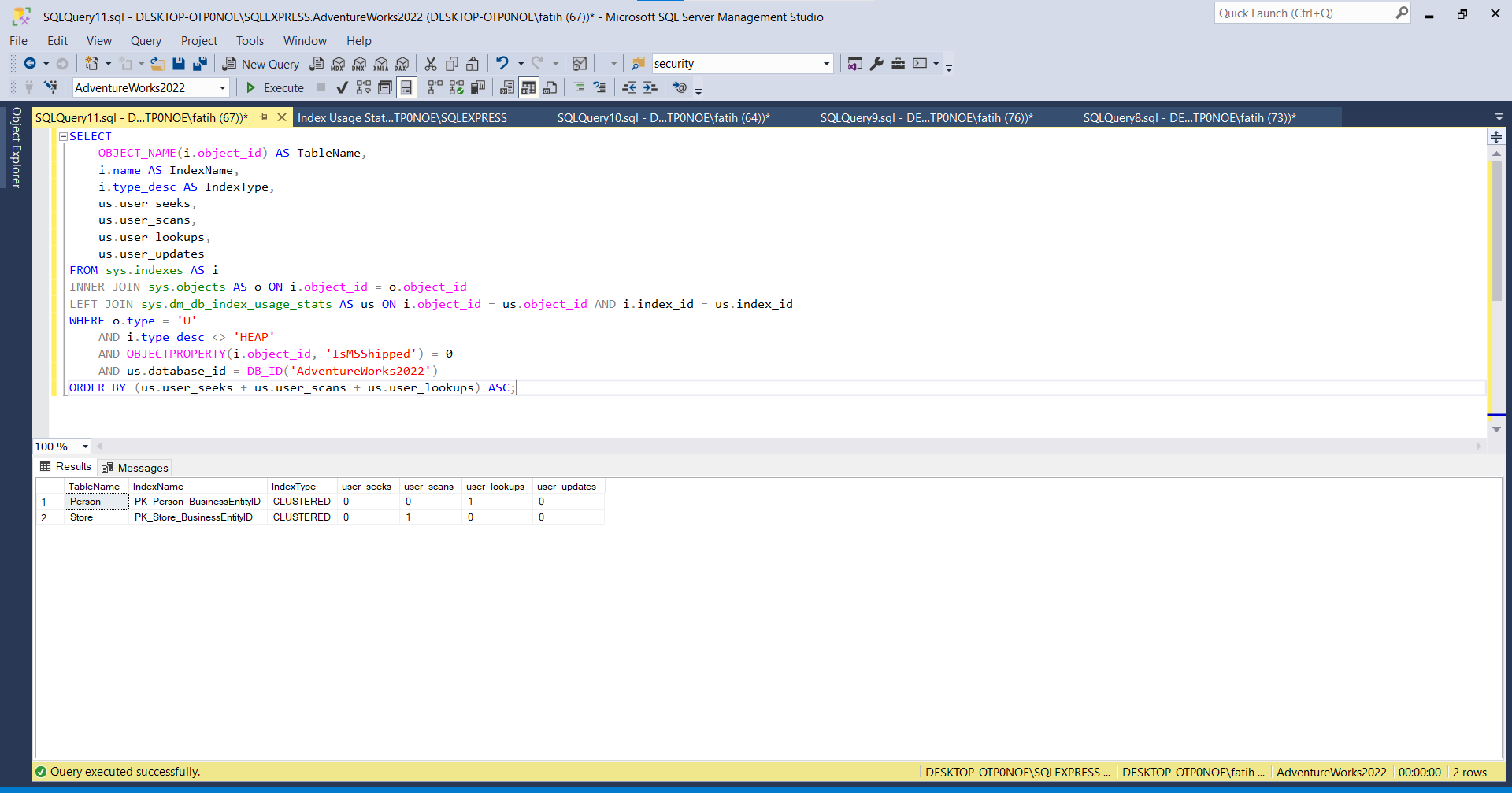
*WHERE o.type = 'U'*

*AND i.type\_desc <> 'HEAP'*

*AND OBJECTPROPERTY(i.object\_id, 'IsMSShipped') = 0*

*AND us.database\_id = DB\_ID('AdventureWorks2022')*

*ORDER BY (us.user\_seeks + us.user\_scans + us.user\_lookups) ASC;*

****

**c. Sorgu İyileştirme**

Execution Plan ile Sorgu Analizi için;

1. İncelemek istenen sorgu yazılır. Örnek sorgu aşağıda gösterilmiştir.
2. Include Actual Execution Plan seçeneğini açılır (Ctrl+M).
3. Sorgu çalıştırılır.
4. Execution Plan sekmesinde darboğazlar kontrol edilir.

Örnek sorgu:

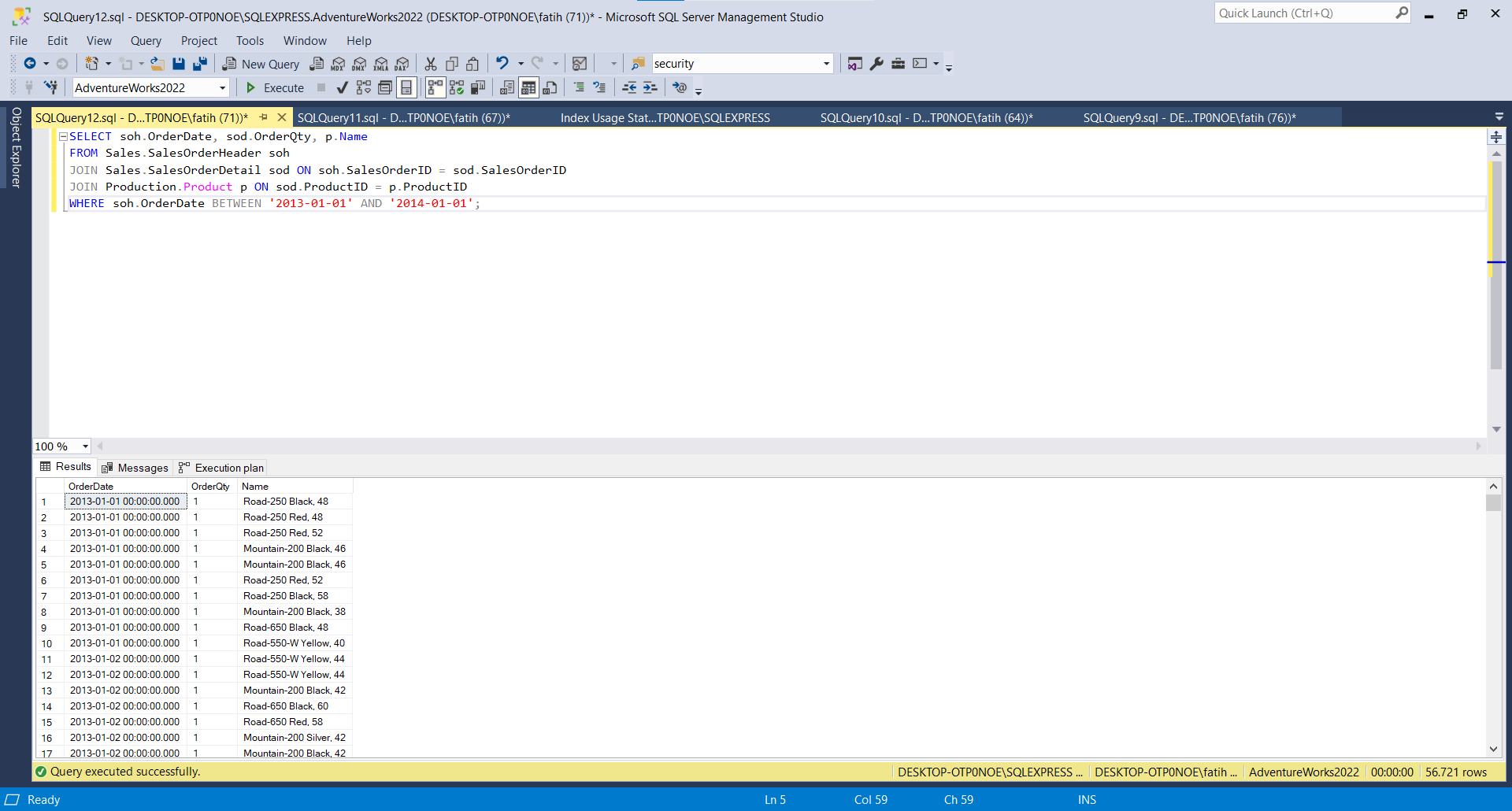
*SELECT soh.OrderDate, sod.OrderQty, p.Name*

*FROM Sales.SalesOrderHeader soh*

*JOIN Sales.SalesOrderDetail sod ON soh.SalesOrderID = sod.SalesOrderID*

*JOIN Production.Product p ON sod.ProductID = p.ProductID*

*WHERE soh.OrderDate BETWEEN '2013-01-01' AND '2014-01-01';*

****

**d. Veri Yöneticisi Rolleri**

Yeni Kullanıcı Oluşturma ve Yetkilendirme;

SQL Server’a bağlanabilecek bir kullanıcı tanımlamak için;

1. Sol menüde "Security"> "Logins" üzerine sağ tıklayarak → "New Login…" seçilir.

Açılan pencerede:

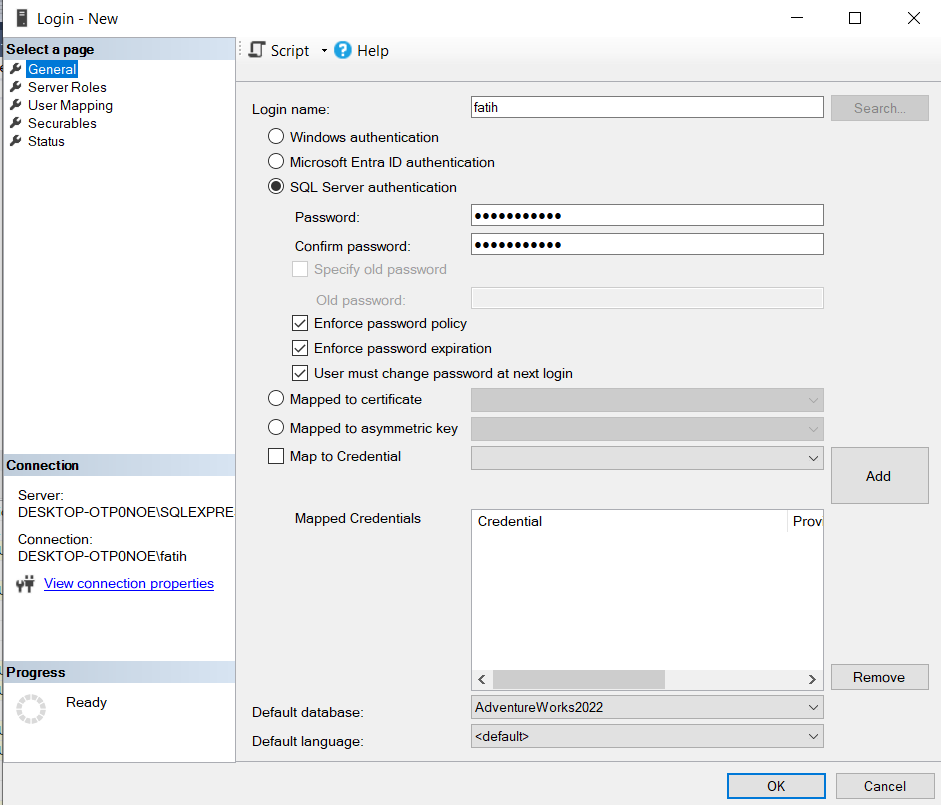
Login name: ‘Fatih’ yazılır.

* SQL Server Authentication seçilir.
* Güçlü bir şifre girilir.
* Enforce password policy kutusu işaretli bırakılır.
* Default database: AdventureWorks2022 olarak ayarlanır.

Sol menüden "Server Roles" sekmesine tıklanır.

public seçili bırakılır (default).

OK diyerek login oluşturulur.

****

Login’i Veritabanında Kullanıcıya Dönüştürmek için;

1. Object Explorer’da Databases> AdventureWorks2022> Security> Users kısmına sağ tıklanır → "New User…"
2. Açılan pencerede:

User name: fatihlimanlik

Login name: sağdaki … tuşuyla az önce oluşturduğumuz login (Fatih) seçilir.

Default Schema: genellikle dbo olarak bırakılır.

1. OK diyerek kullanıcı oluşturulur.

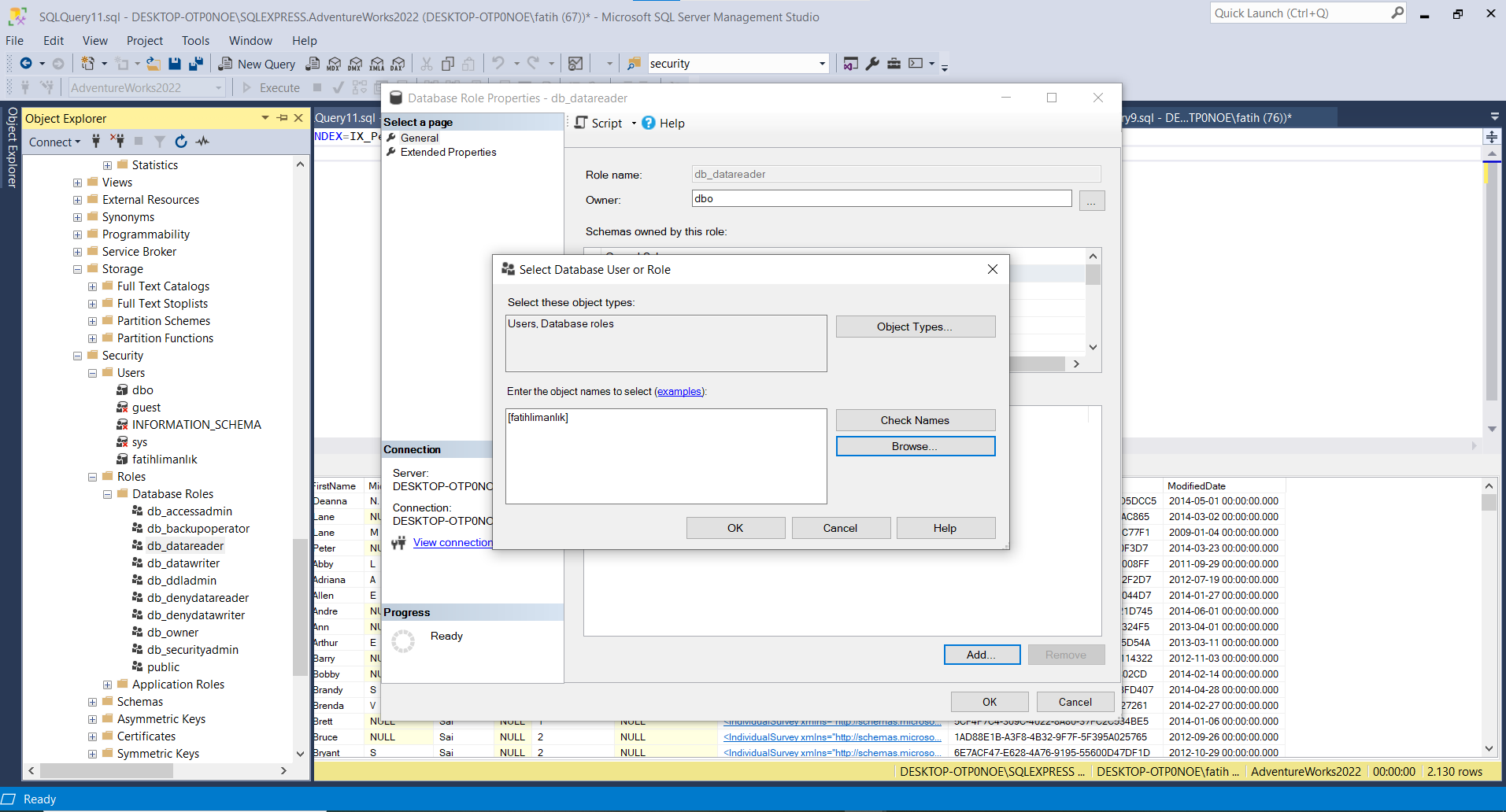
Kullanıcıya yetki(rol) vermek için;

1.Databases> AdventureWorks2022> Security> Roles> Database Roles yolu izlenir.

2.db\_datareader rolüne sağ tıklanır → **"Properties"**

3.Sol alttan **"Add"** butonuna tıklanır, çıkan listeden “fatihlimanlik” seçilir → OK.

Artık bu kullanıcı veri tabanındaki tüm tabloları okuyabilir.



**2. VERITABANI YEDEKLEME VE FELAKETTEN KURTARMA PLANI**

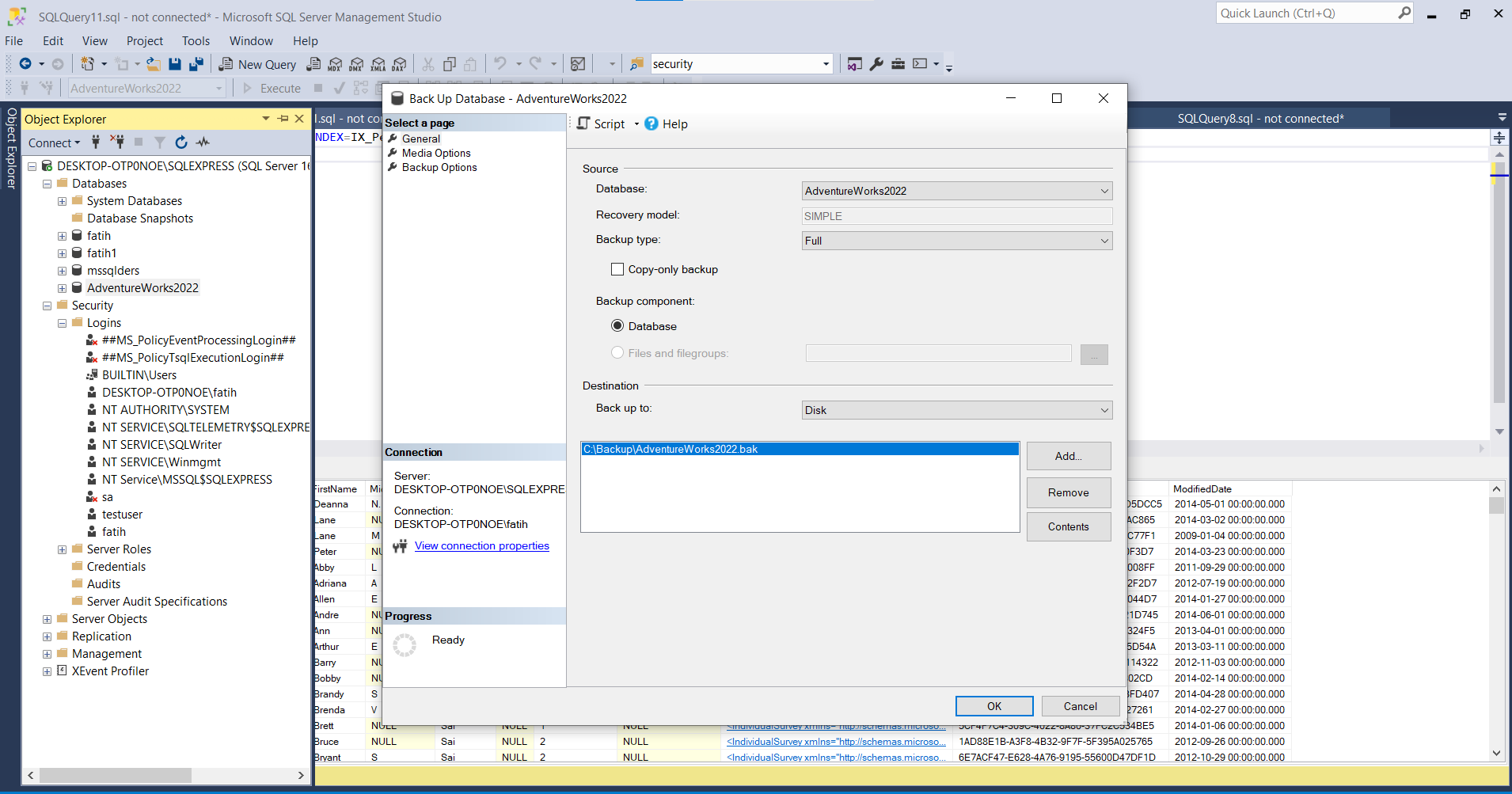
**Tam (Full) Yedek Alma:**

SSMS’te Databases> AdventureWorks2022 üzerine sağ tıklanır. → Tasks> Back Up…

Açılan pencerede:

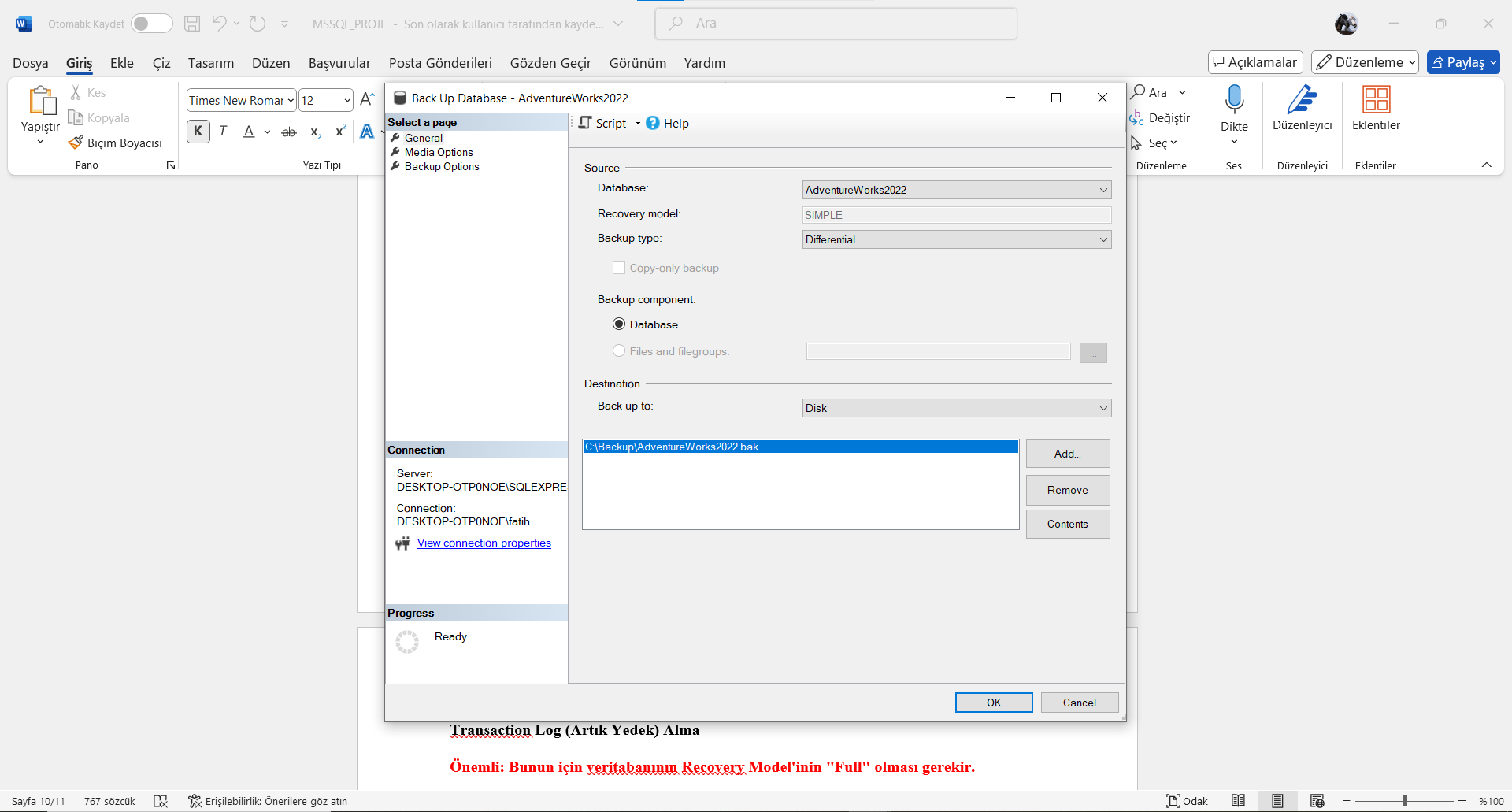
* Backup Type: Full seçilir.
* Destination: Varsayılan yolu görülecektir. Remove ile silinip Add tuşuyla farklı bir konum belirlenir (C:\Backups\AdventureWork\_full\_backup.bak ).

2.OK tıklanarak tam yedek alınır.

****

**Fark (Differential) Yedek Alma:**

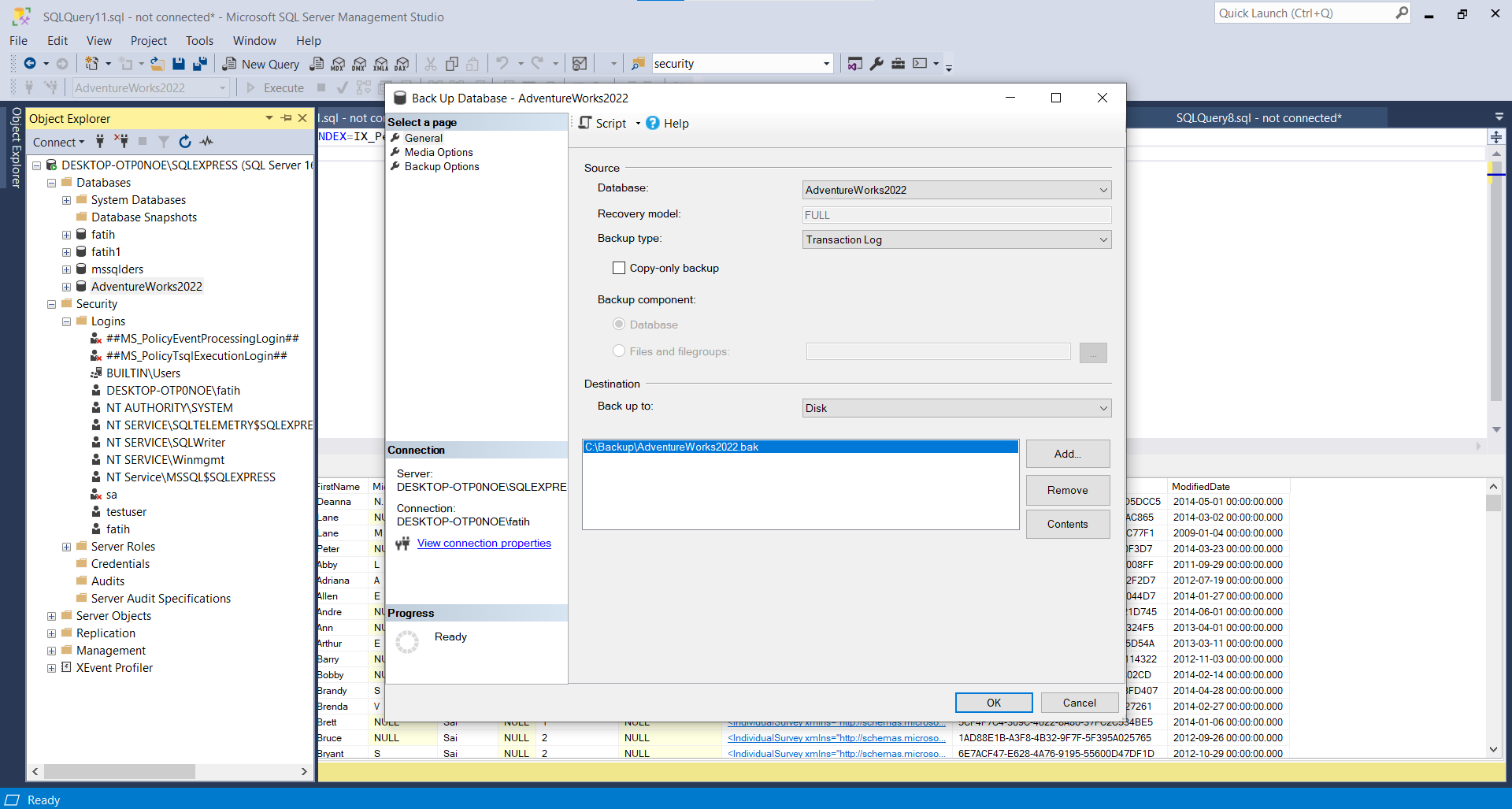
1. Databases> AdventureWorks2022 sağ tıklanır. → Tasks> Back Up…
2. Backup Type: Differential seçililir.
3. Aynı veya farklı bir .bak dosyası olarak kaydedilir.
4. OK tıklayarak fark yedeği alınır.

****

**Transaction Log (Artık Yedek) Alma**

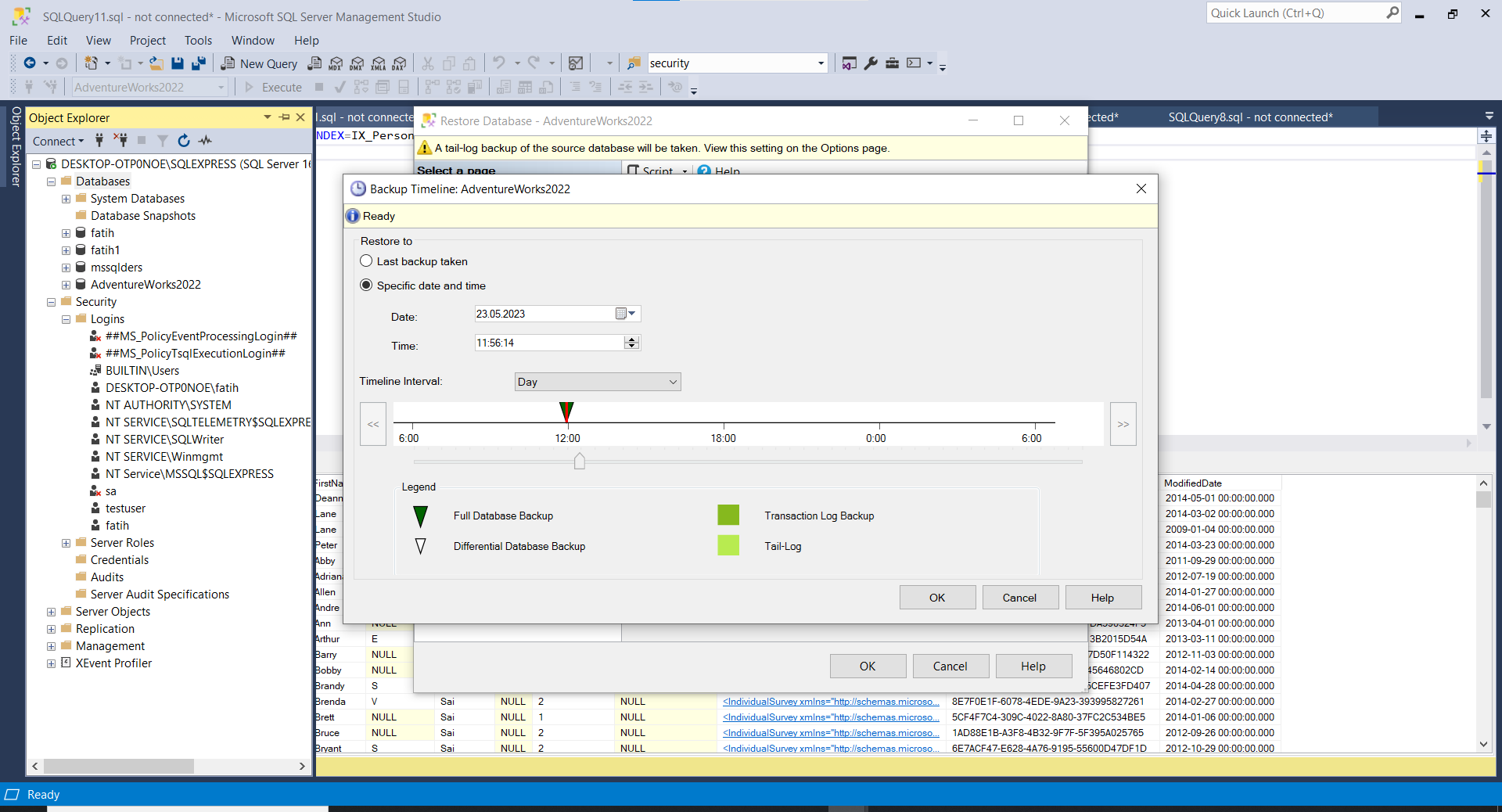
Bunun için veritabanının Recovery Model'inin "Full" olması gerekir.

1. Databases> AdventureWorks2022 sağ tıklanır → Properties> Options
2. Recovery model: Full olarak değiştirilir.
3. Tekrar Tasks> Back Up… yolu izlenir.
4. Backup Type: Transaction Log seçilir.
5. Lokasyonu ayarlanır → OK

****

**Point-in-Time Restore (Belirli Bir Zamana Geri Yükleme)**

1. Databases klasörüne sağ tıklanır ve Restore> Database... seçeneği seçilir.
2. Source bölümünde "Device" seçeneğini seçilir ve "..." butonu tıklanır.
3. Timeline butonuna tıklanarak "Specific date and time" seçeneğiyle geri yüklenecek spesifik zamanı seçilir.
4. Files sekmesinde dosyaların restore edileceği konumlar kontrol edilir.
5. Options sekmesinde "Overwrite the existing database" seçeneği işaretlenir. "Recovery state" kısmında "RESTORE WITH RECOVERY" seçeneği seçilir.
6. OK butonu tıklanarak geri yükleme işlemi başlatılır.

****

**Felaketten Kurtarma Senaryosu**

Person.Person tablosunun yanlışlıkla silindiğini varsayalım. Önceden aldığımız tam yedeği geri yükleriz:

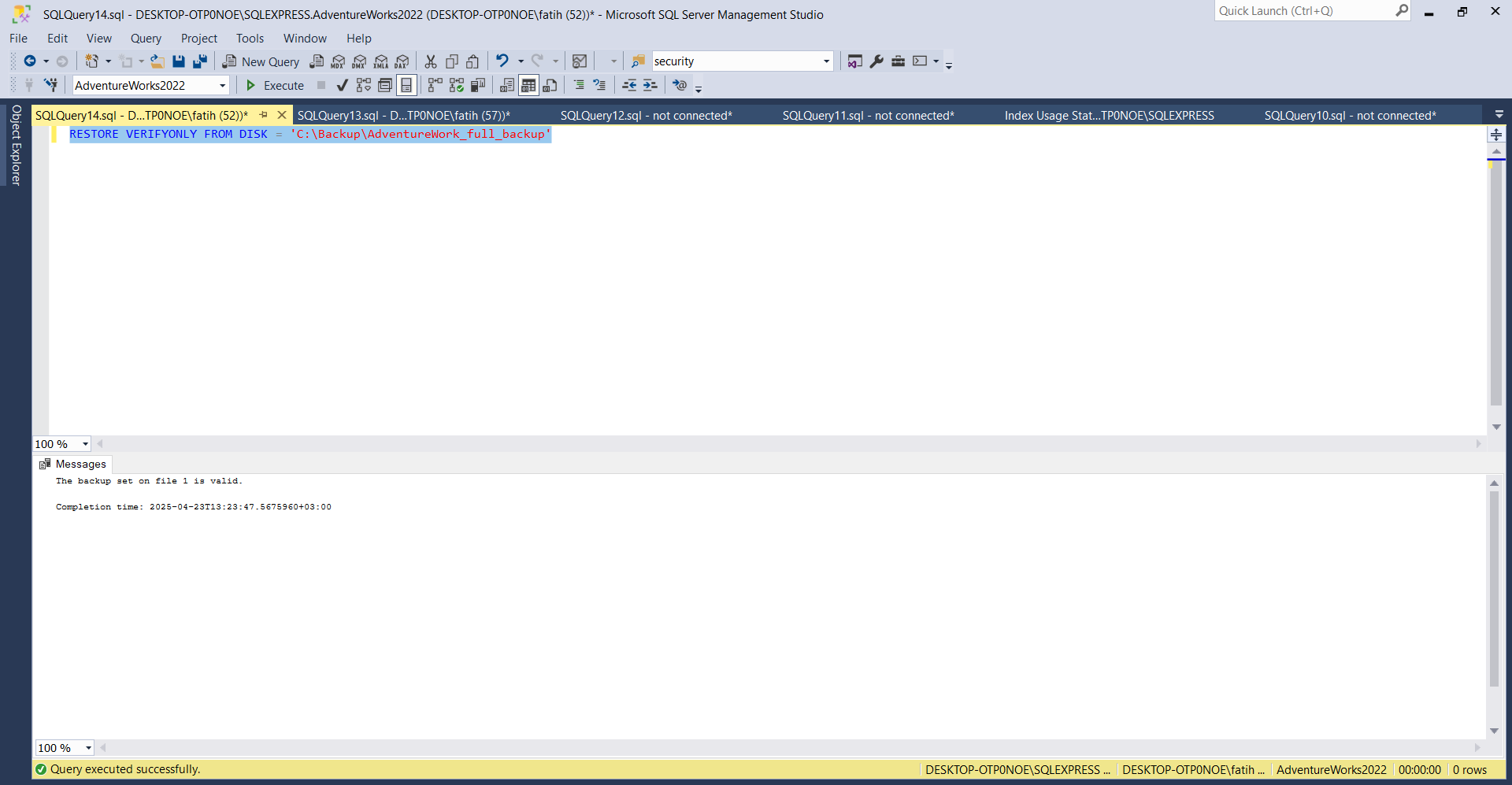
1. Databases> AdventureWorks2022 sağ tıklanır → Tasks> Restore> Database…
2. Source: Device seçilir→ .bak dosyası eklenir.
3. Destination: AdventureWorks2022 olarak seçilir.
4. Alt menüde Options sekmesine geçilir: Overwrite the existing database kutusu işaretlenir.
5. OK diyerek yedek geri yüklenir.
6. Eğer sadece tablo silindiyse ve tüm veritabanını geri yüklemek istenmiyorsa, farklı bir veritabanı adıyla yedek açılıp oradan gerekli tablo taşınır.

**Test Yedekleme Senaryoları**

Yedek geri yüklenebilir mi test etmek için SSMS'de yeni bir sorgu penceresi açılarak şaşağıdaki komut çalıştırılır.

*RESTORE VERIFYONLY*

*FROM DISK = 'C:\Backup\AdventureWork\_full\_backup'*



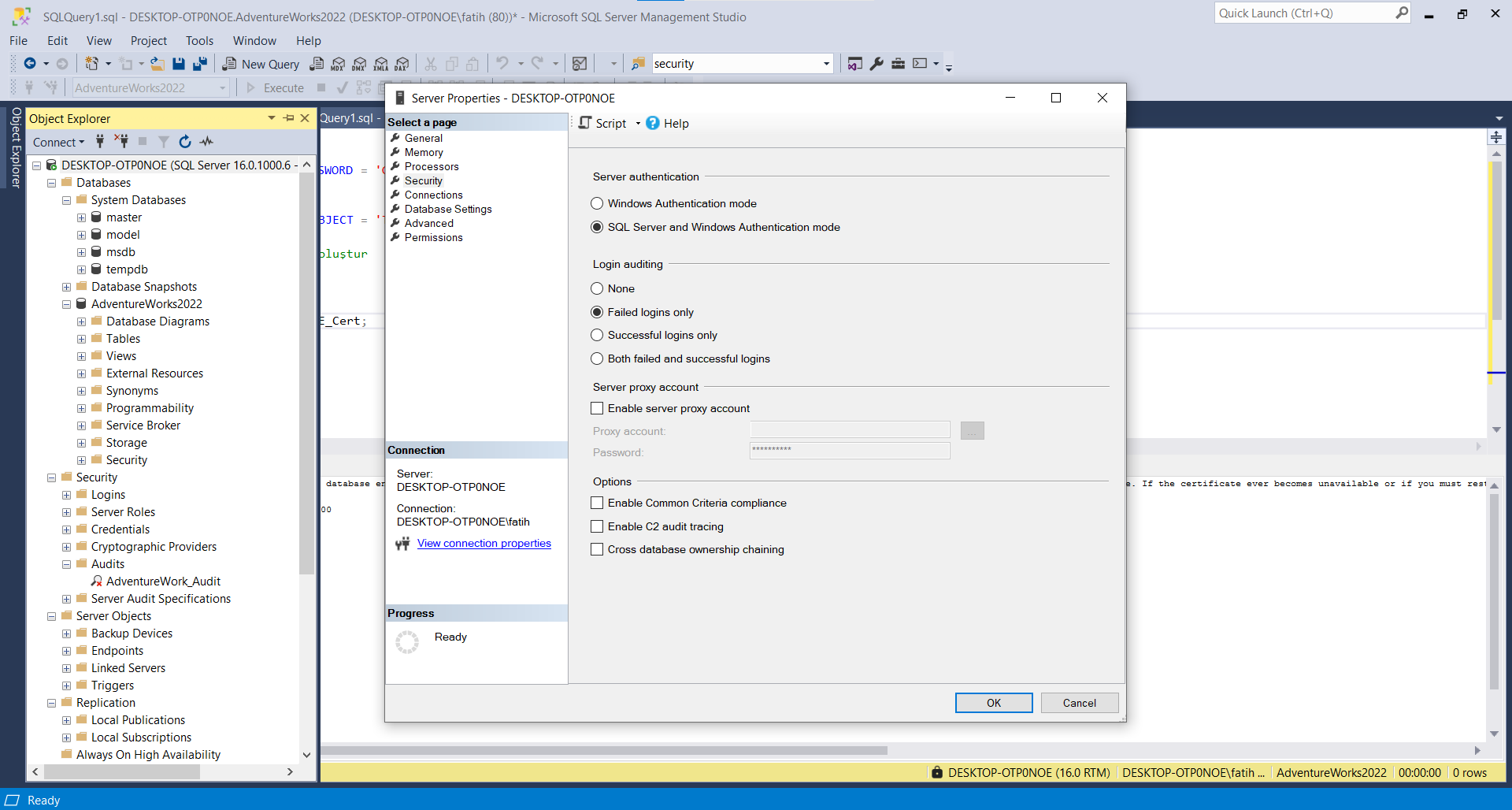
**3.VERİTABANI GÜVENLİĞİ VE ERİŞİM KONTROLÜ**

SQL Server ve Windows Authentication için;

"Security" sayfası seçilir ve "Server authentication" bölümünden authentication modu ayarlanır:

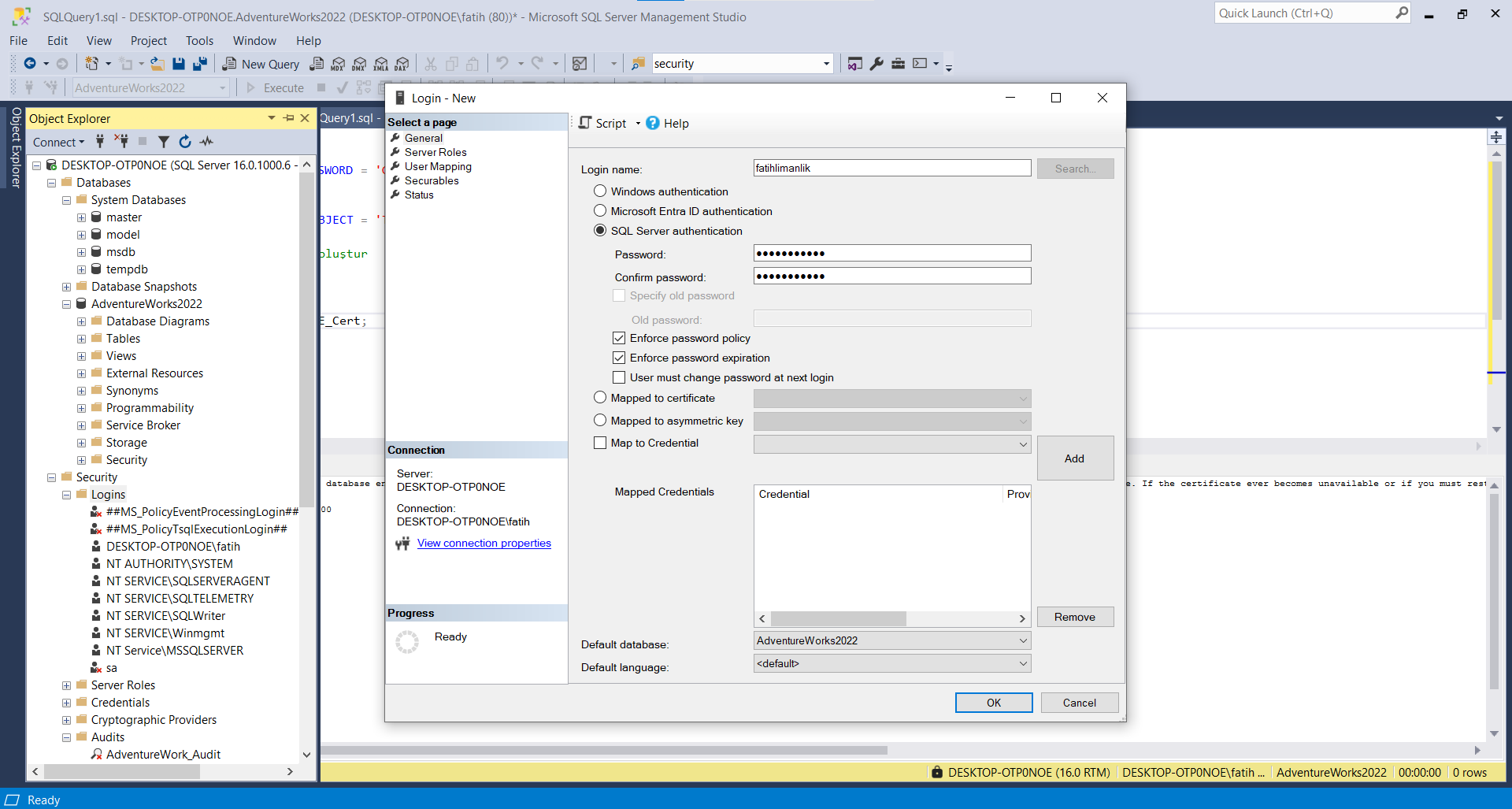
"Windows Authentication mode" yalnızca Windows kimlik doğrulaması kullanır.

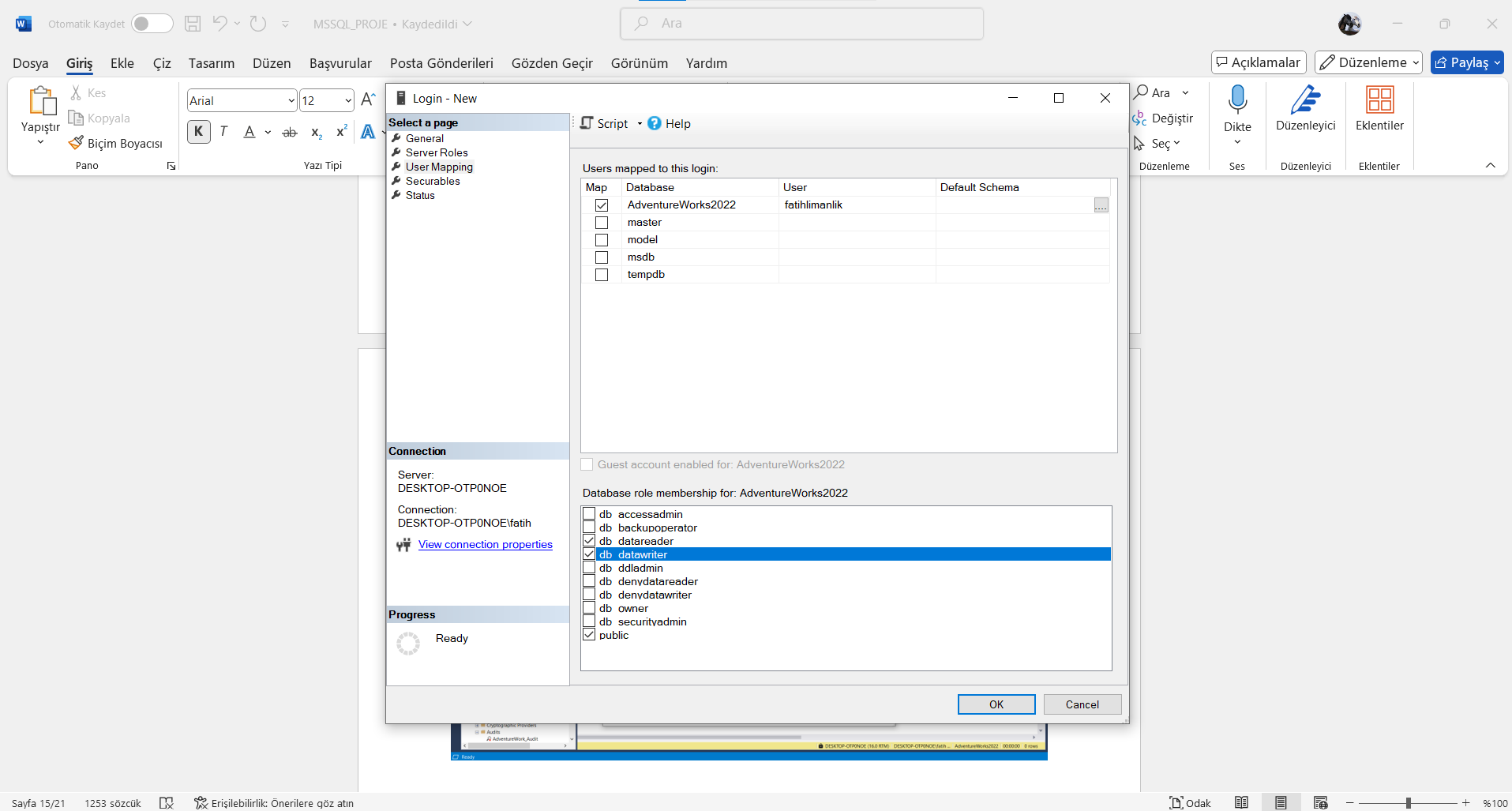
"SQL Server and Windows Authentication mode" her ikisinin de kullanılabilmesini sağlar.

****

**Yeni SQL Server Kullanıcısı Oluşturma**

1. Object Explorer'da "Security"> "Logins" klasörüne sağ tıklanır ve "New Login" seçilir.
2. Kullanıcı adı girilir ve "SQL Server authentication" seçeneği işaretlenir.
3. Güçlü bir şifre belirlenir ve "Enforce password policy" opsiyonu aktif edilir.
4. "User Mapping" sayfasında AdventureWorks2022 veritabanı seçilir ve bu kullanıcının sahip olacağı roller (db\_datareader, db\_datawriter vb.) belirlenir.

****

****

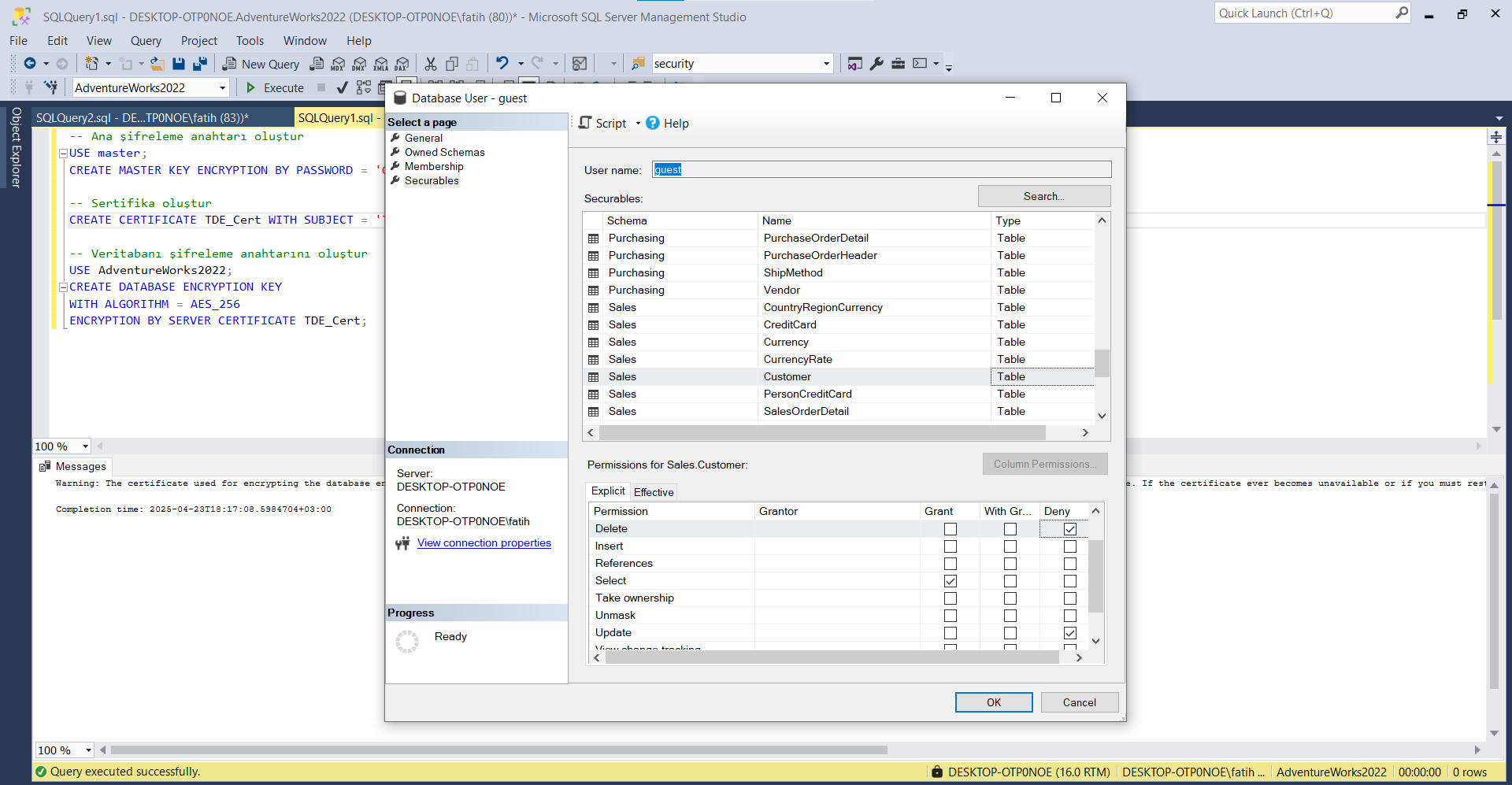
**Veritabanı Rollerine Erişim Tanımlama**

Örneğin, sadece Sales tablosuna erişim vermek için:

AdventureWorks2022> Security> Users altında ilgili kullanıcıya sağ tıklanır ve "Properties" seçilir.

"Securables" sekmesinde "Search" butonu ile "Sales.Customer" tablosu seçilir.

SELECT izni verilir fakat UPDATE veya DELETE izni verilmez.

****

**Veri Şifreleme (TDE Sertifikası Oluşturma)**

1. AdventureWorks2022 veritabanı açılır ve aşağıdaki sorgu çalıştırılır:

*-- Ana şifreleme anahtarı oluşturmak için*

*USE master;*

*CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'GucluSifre123!';*

*-- Sertifika oluşturmak için*

*CREATE CERTIFICATE TDE\_Cert WITH SUBJECT = 'TDE Certificate for AdventureWorks2022';*

*-- Veritabanı şifreleme anahtarını oluşturmak için*

*USE AdventureWorks2022;*

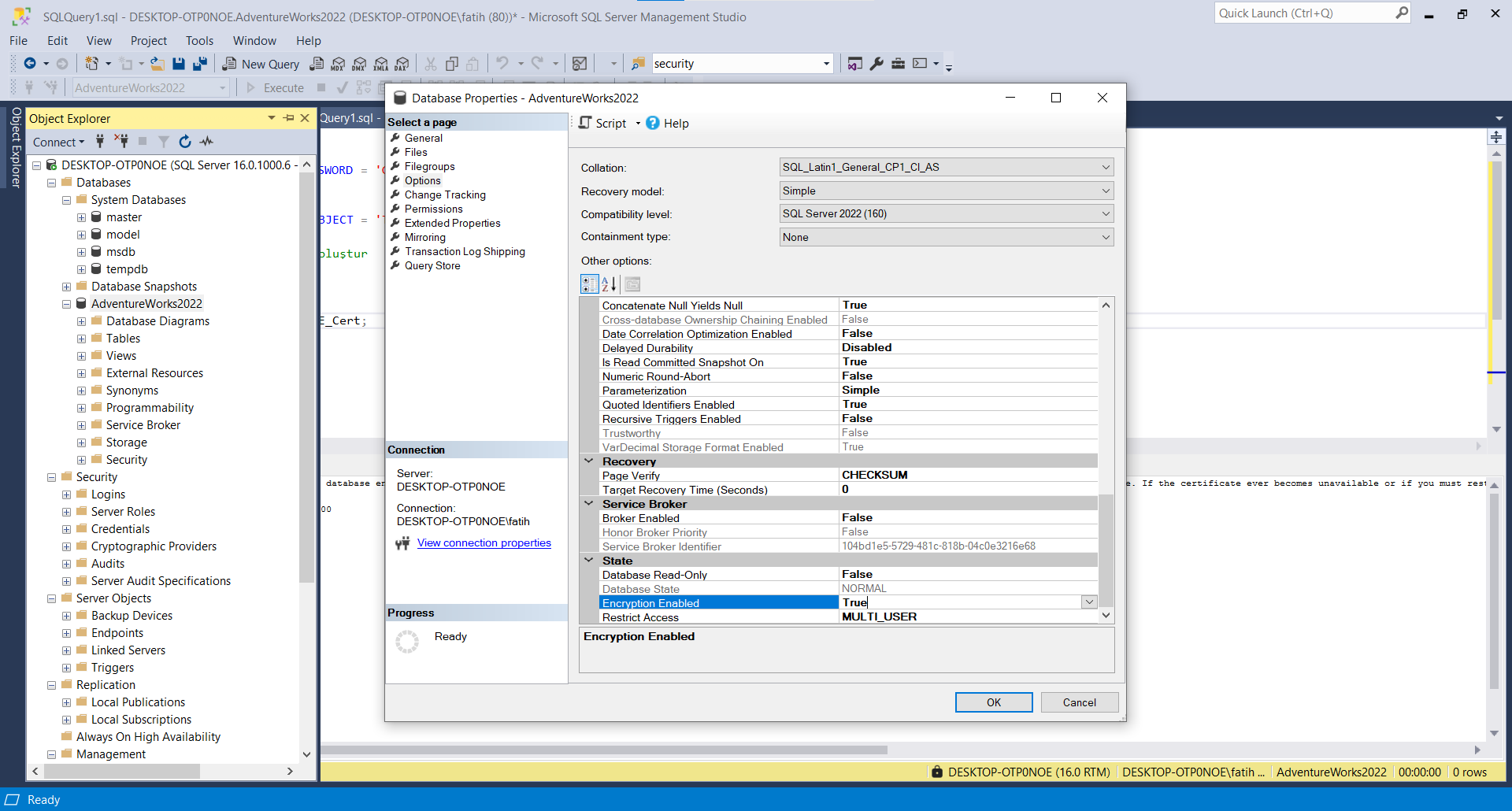
*CREATE DATABASE ENCRYPTION KEY*

*WITH ALGORITHM = AES\_256*

*ENCRYPTION BY SERVER CERTIFICATE TDE\_Cert;*

**TDE Etkinleştirme**

1. AdventureWorks2022 veritabanına sağ tıklanır ve "Properties" seçilir.
2. "Options" sayfasında "Encryption Enabled" özelliği "True" olarak ayarlanır.
3. "OK" düğmesine tıklanır ve işlem onaylanır.



Şifreleme Durumunu Kontrol Etmek İçin aşağıdaki sorgu çalıştırılır.

*SELECT DB\_NAME(database\_id) AS DatabaseName,*

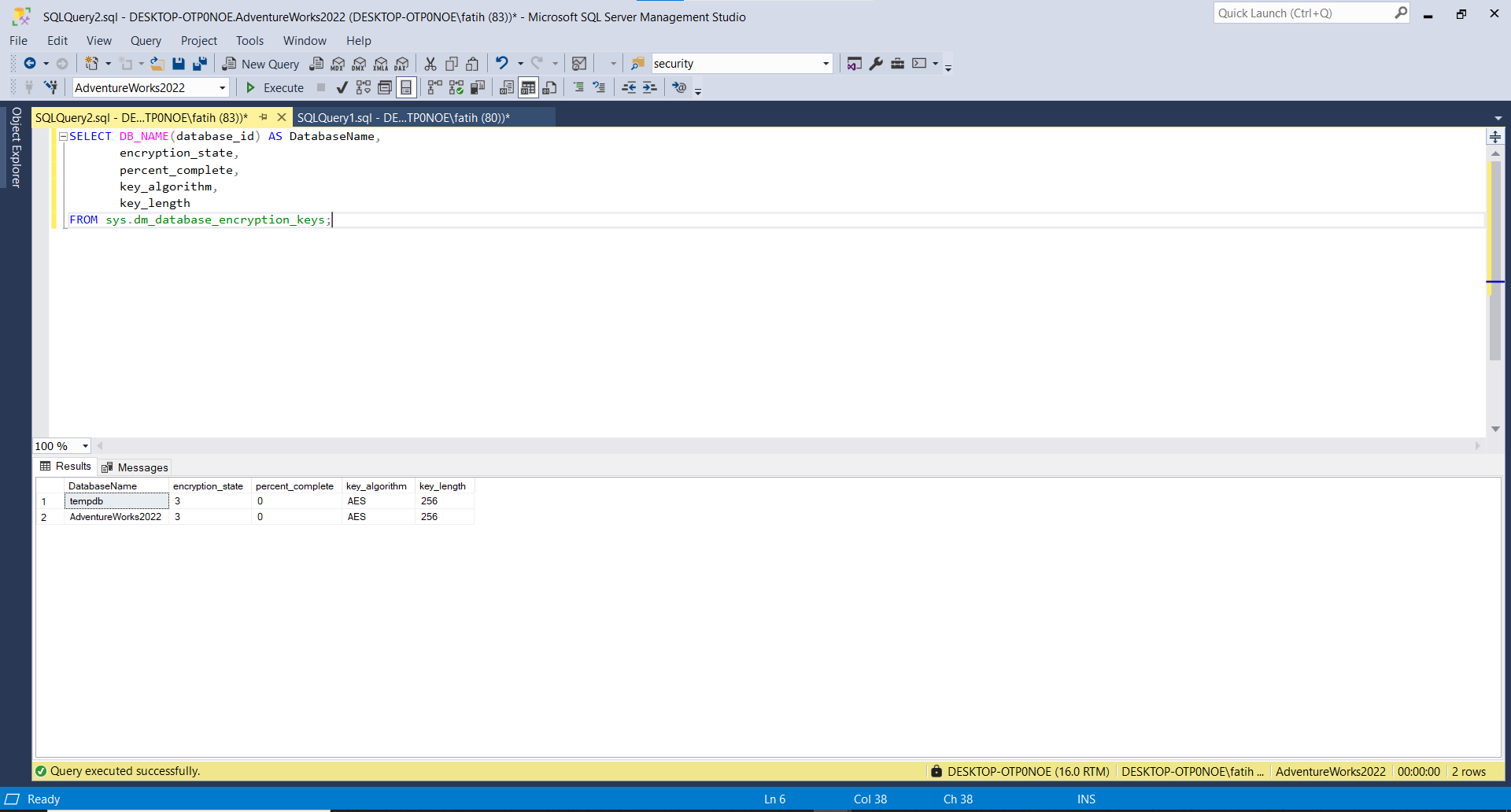
*encryption\_state,*

*percent\_complete,*

*key\_algorithm,*

*key\_length*

*FROM sys.*dm*\_database\_encryption\_keys;*



**sql injection testleri ve önlemleri;**

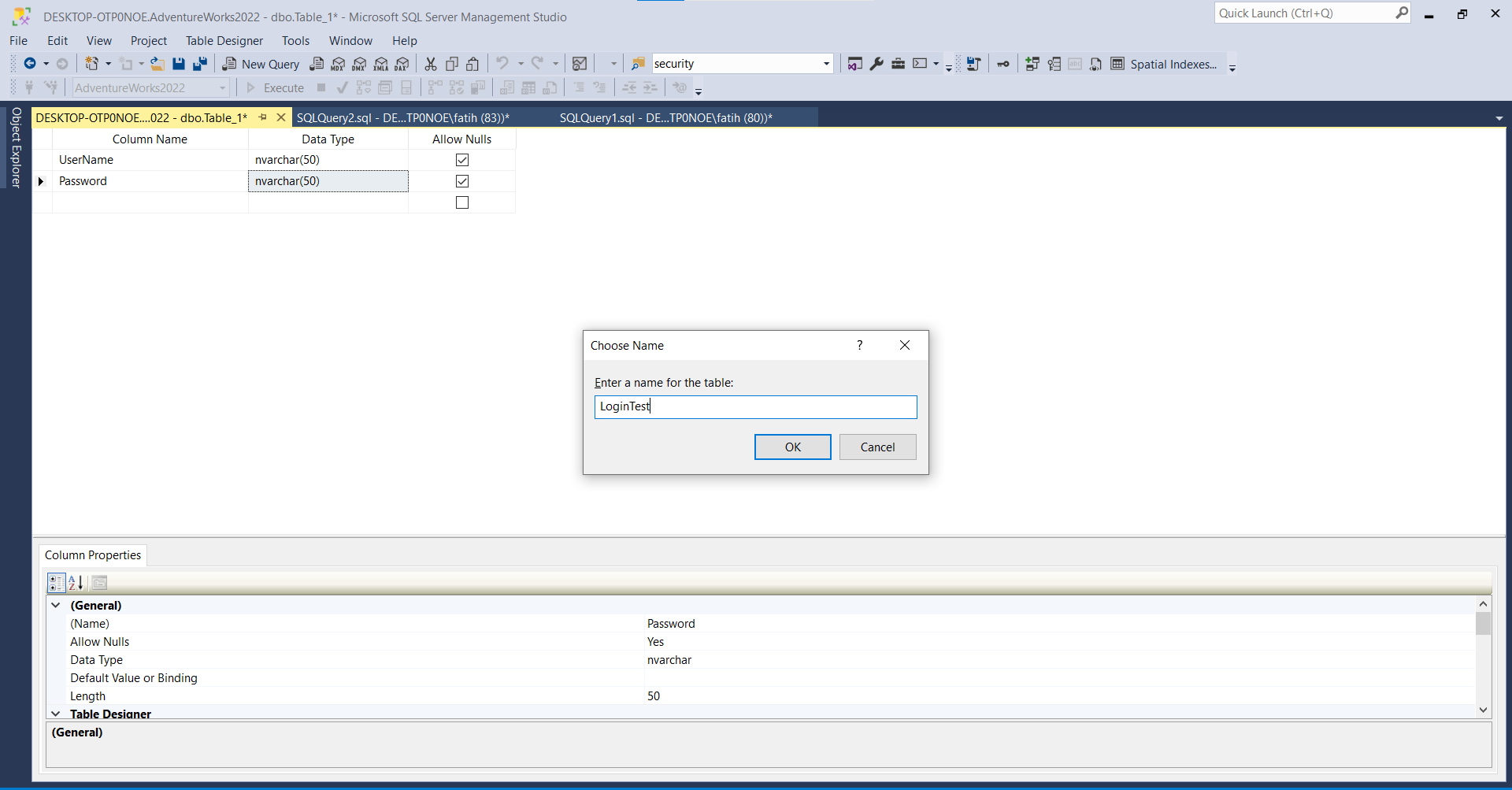
AdventureWorks2022’de hazır stored procedure’ler doğrudan injection açıklığı içermez. Bu yüzden örnek bir açık prosedür oluşturmak gerekir.

Örnek Açığa Uygun Tablonun Oluşturulması

1. AdventureWorks2022> Tables klasörüne sağ tıklanır → New Table
2. Aşağıdaki sütunlar eklenir:

| Column Name | Data Type |
| --- | --- |
| UserName | nvarchar(50) |
| Password | nvarchar(50) |

Tablo adını “LoginTest” olarak kaydedilir.

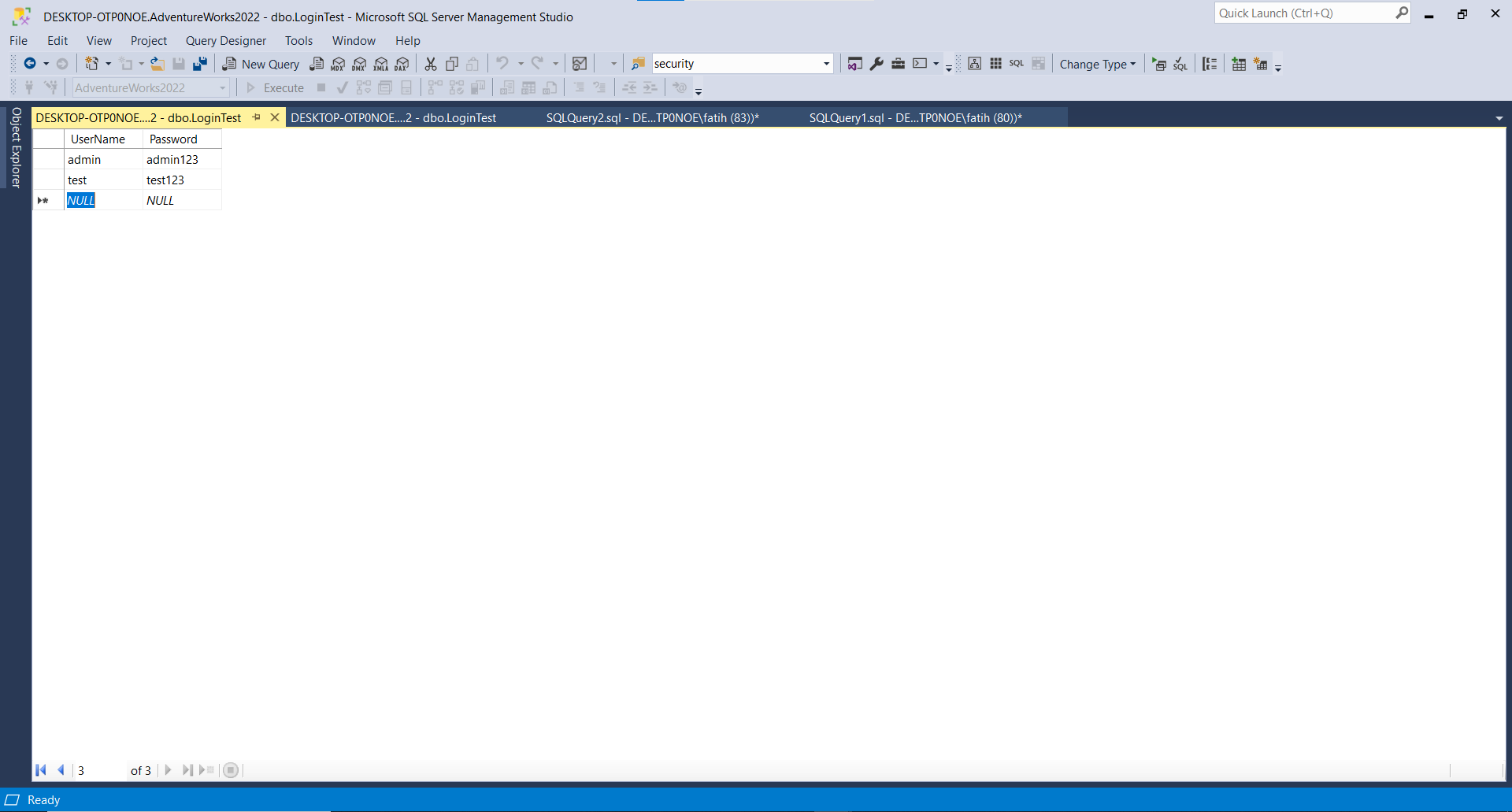


**Test verisi ekleme**

1. Tables> LoginTest sağ tıklanır→ Edit Top 200 Rows
2. Örnek veri girilir:

UserName: admin, Password: admin123

UserName: test, Password: test123



**SQL Injection Açığı İçeren Stored Procedure Oluşturma;**

1. Programmability> Stored Procedures sağ tıklanır → New Stored Procedure
2. Aşağıdaki kod girilir:

*CREATE PROCEDURE dbo.CheckLoginInjection*

*@Username NVARCHAR(50),*

*@Password NVARCHAR(50)*

*AS*

*BEGIN*

*DECLARE @sql NVARCHAR(MAX)*

*SET @sql = 'SELECT \* FROM LoginTest WHERE UserName = ''' + @Username + ''' AND Password = ''' + @Password + ''''*

*EXEC (@sql)*

*END*

Prosedür adı “CheckLoginInjection” olarak kaydedilir.

**SQL Injection Saldırısını Test Etme;**

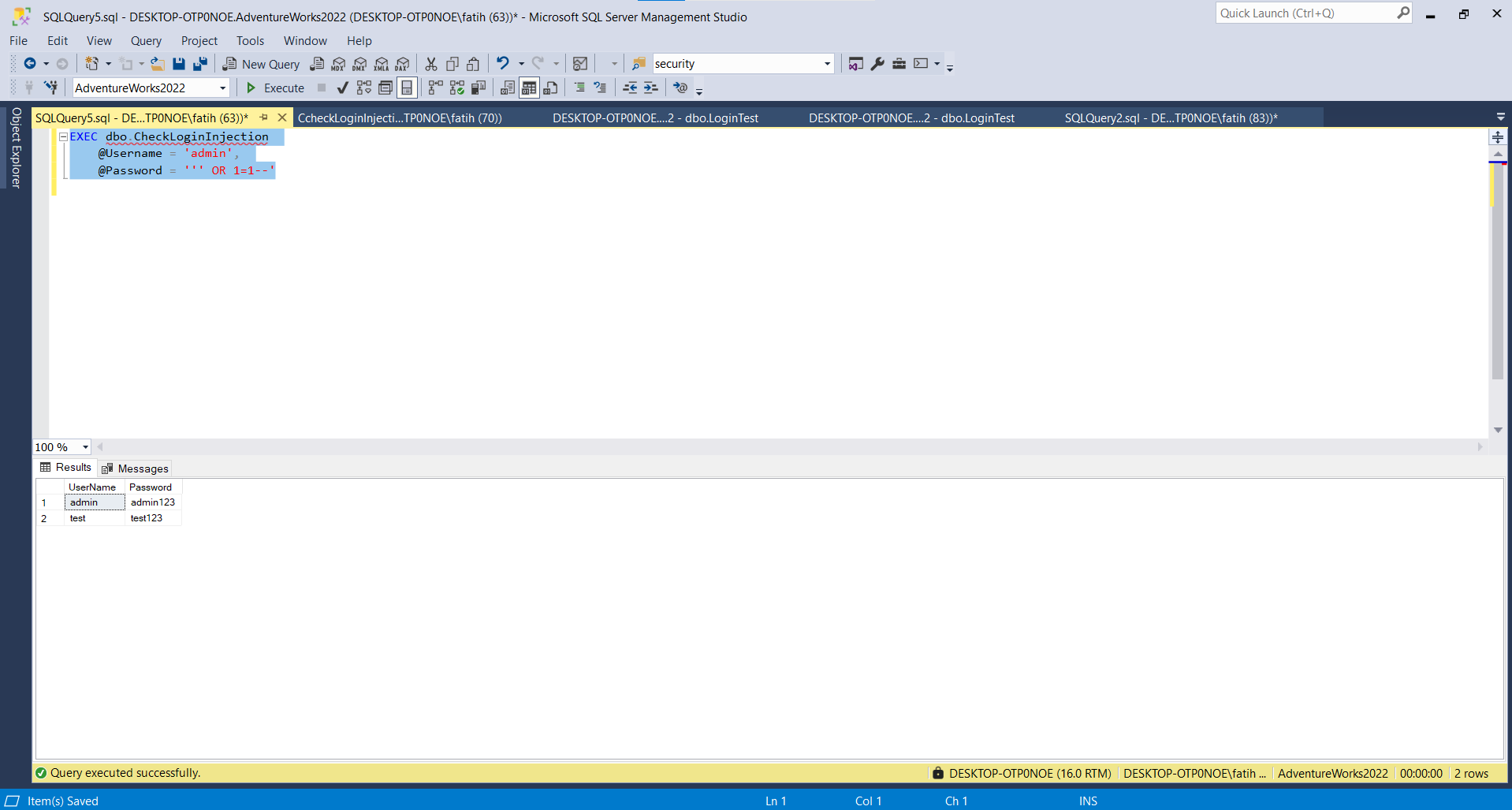
1. New Query açılır.
2. Şu sorgu yazılır ve çalıştırılır:

*EXEC dbo.CheckLoginInjection*

*@Username = 'admin',*

*@Password = ''' OR 1=1--'*

Bu sorgu, WHERE koşulunu devre dışı bırakarak tüm kullanıcıları getirir. Injection başarılı olmuştur.



**SQL Injection Açığını Güvenli Hale Getirme;**

GUI üzerinden Stored Procedure> dbo.CheckLoginInjection → sağ tıklanır → Modify → aşağıdaki şekilde değiştirilir:

*ALTER PROCEDURE dbo.CheckLoginInjection*

*@Username NVARCHAR(50),*

*@Password NVARCHAR(50)*

*AS*

*BEGIN*

*SELECT \* FROM LoginTest*

*WHERE UserName = @Username AND Password = @Password*

*END*

Bu haliyle, parametrik sorgu kullanıldığından injection girişimleri başarısız olur.

**SQL Injection Denemesi (Güvenli Sürümde)**

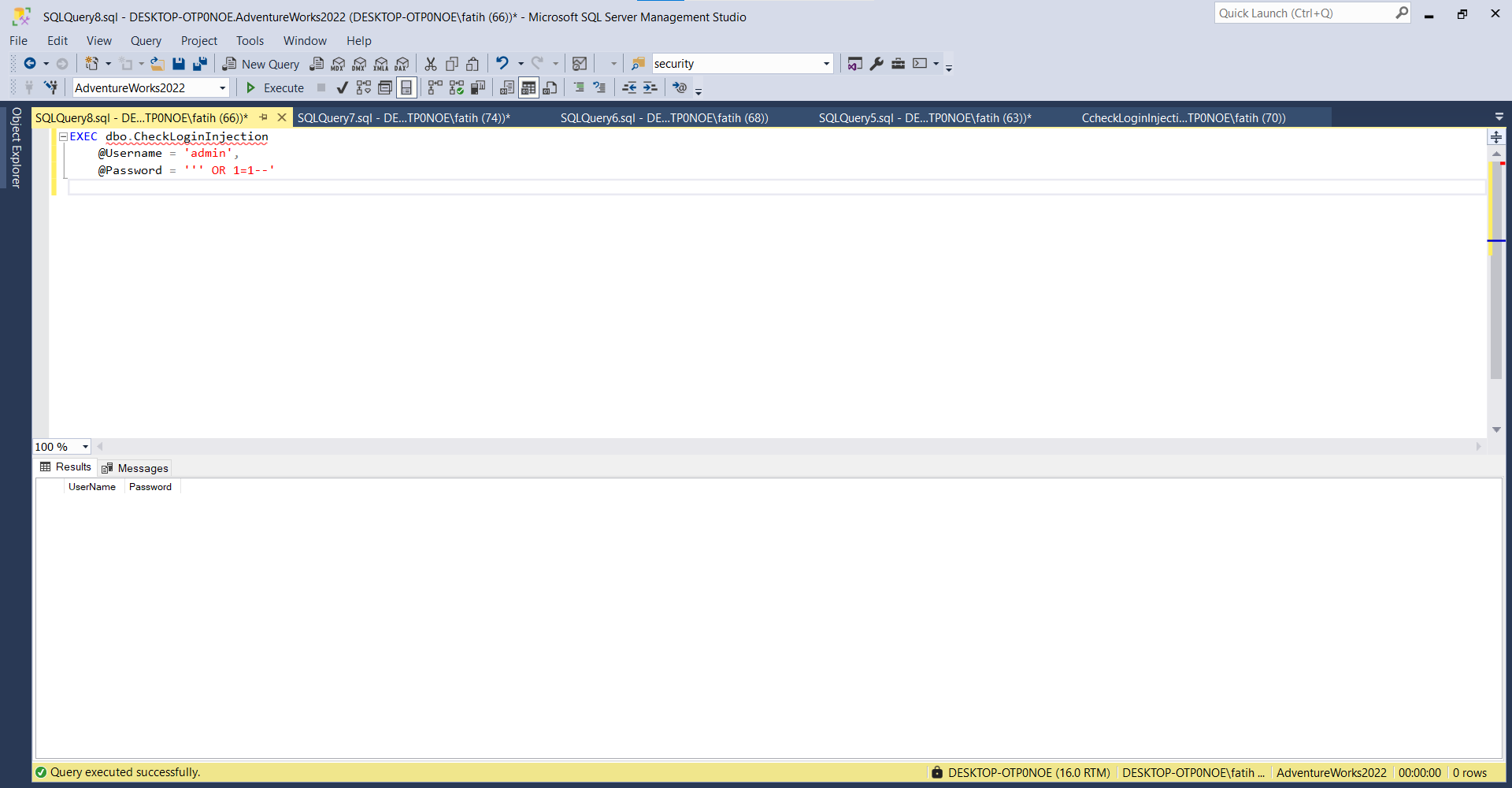
Tekrar aşağıdaki sorgu çalıştırılır:

*EXEC dbo.CheckLoginInjection*

*@Username = 'admin',*

*@Password = ''' OR 1=1--'*

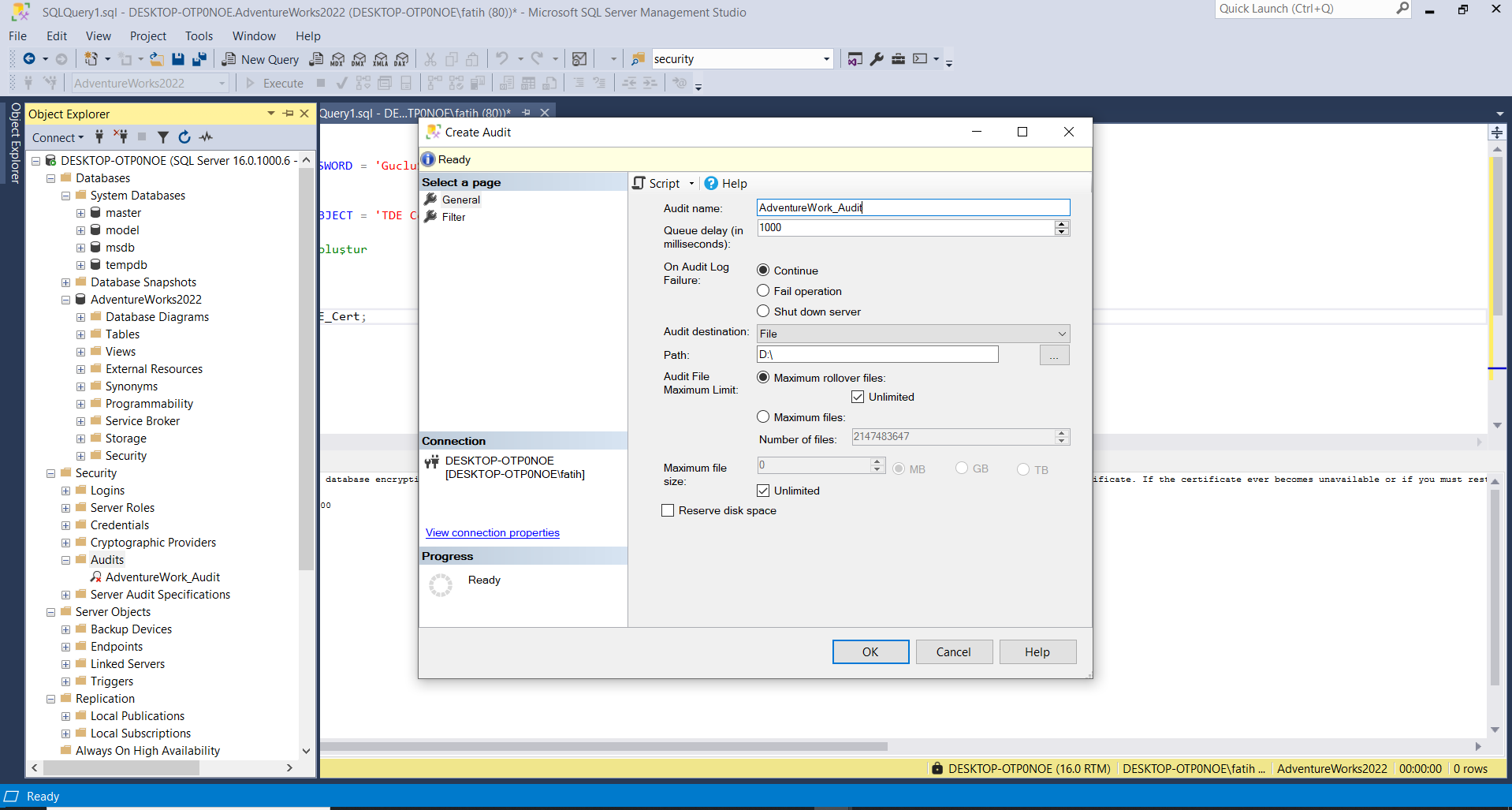
Artık hiçbir kullanıcı dönmez. Çünkü bu, artık zararsız bir parametredir.



**Audit Logları Oluşturma ve Yönetme**

**SQL Server Audit Oluşturma;**

1. Object Explorer'da "Security"> "Audits" klasörüne sağ tıklanır.
2. "New Audit" seçeneğini seçilir.
3. Audit için bir isim verilir ("AdventureWorks\_Audit").
4. "File" destination type seçilir ve dosya yolu belirlenir, "OK" düğmesine tıklanır.



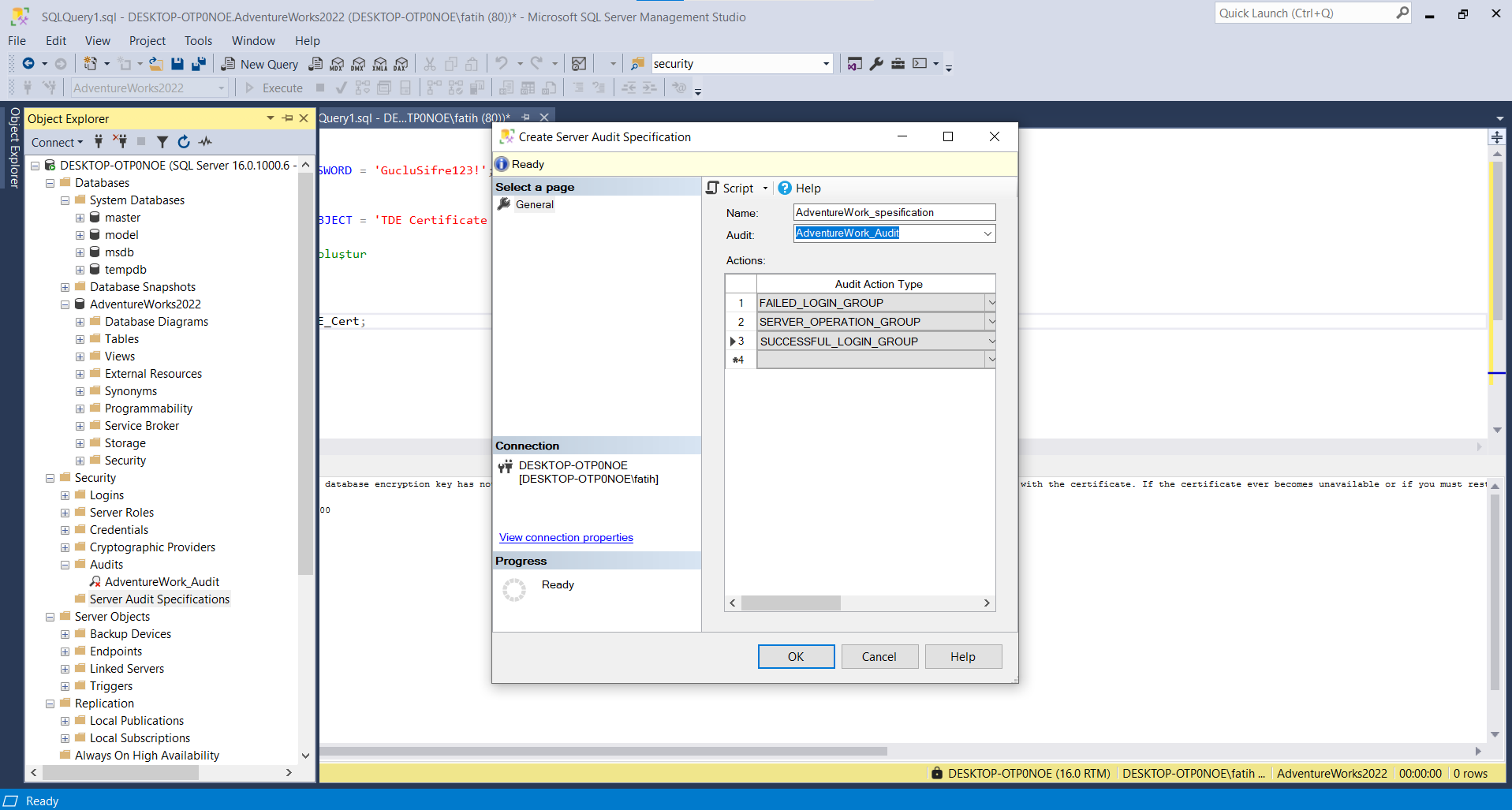
**Sunucu Audit Spesifikasyonu Oluşturma**

1. "Security"> "Server Audit Specifications" klasörüne sağ tıklanır.
2. "New Server Audit Specification" seçilir.
3. Bir isim verilir ve oluşturulan “AdventureWork\_Audit” seçilir.
4. "Audit Action Type" bölümünde izlemek istenen eylemler eklenir:

FAILED\_LOGIN\_GROUP: Başarısız oturum açma girişimlerini izler

SUCCESSFUL\_LOGIN\_GROUP: Başarılı oturumları izler

SERVER\_OPERATION\_GROUP: Sunucu yapılandırma değişikliklerini izler



**Audit Loglarını İnceleme**

1. Oluşturulan AdventureWork\_Audit'e sağ tıklanır ve "View Audit Logs" seçilir.
2. Tarih aralığı, olay türü veya kullanıcı gibi filtreleme seçeneklerini kullanarak loglar analiz edilir.

