

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM4522 – Ağ Tabanlı Paralel Dağıtım Sistemleri Vize Proje Raporu**

**Ali Furkan Şahin - 20290289**

**Özge Çelik - 21290514**

<https://github.com/FurkanSahinnn/BLM4522-Ag-tabanli-paralel-dagitim-sistemleri.git>

[https://drive.google.com/file/d/1SiM6hb5KWbFXa\\_GZDMnCZKZEwPmzjuRR/view?usp=drive link](https://drive.google.com/file/d/1SiM6hb5KWbFXa_GZDMnCZKZEwPmzjuRR/view?usp=drive_link)

**25.04.2025**

# **İçindekiler:**

## **1. Veritabanı Performans Optimizasyonu ve İzleme**

- a. Veritabanı İzleme
- b. İndeks Yönetimi
- c. Sorgu İyileştirme
- d. Veri Yöneticisi Roller

## **2. Veritabanı Yedekleme ve Felaketten Kurtarma Planı**

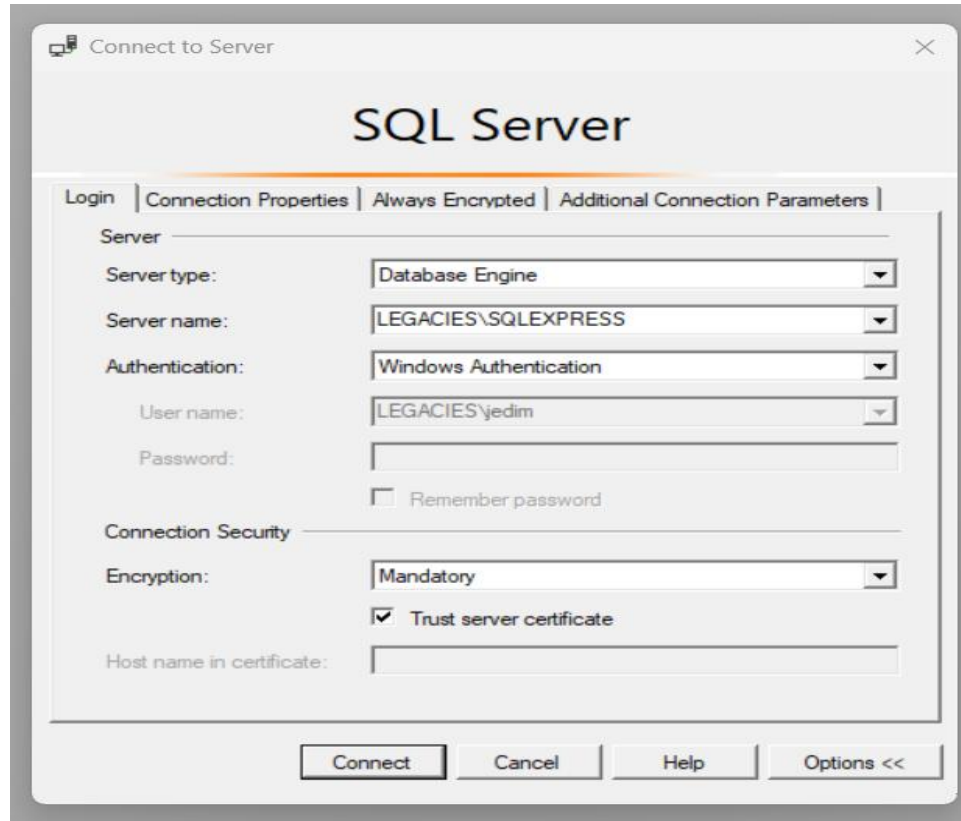
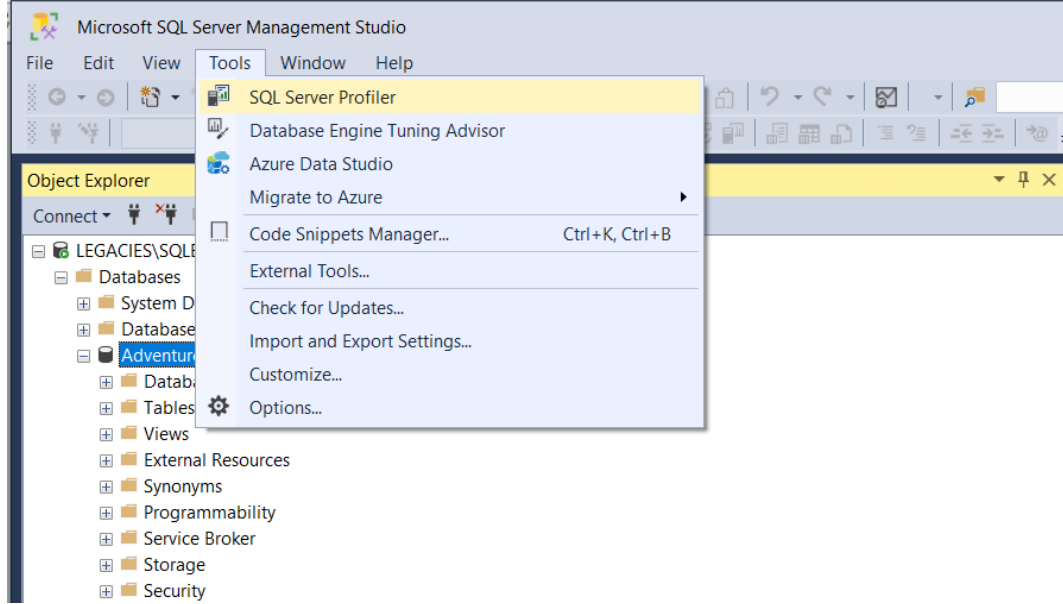
- a. Tam, Artık, Fark Yedeklemeleri
- b. Zamanlayıcılarla Yedekleme
- c. Felaketten Kurtarma Senaryoları
- d. Test Yedekleme Senaryoları

## **3. Veritabanı Güvenliği ve Erişim Kontrolü**

- a. Erişim Yönetimi
- b. SQL Server Authentication ve Windows Authentication Kullanımı
- c. Veri Şifreleme
- d. Audit Logları

# 1. Veritabanı Performans Optimizasyonu ve İzleme

## a. Veritabanı İzleme:



Trace Properties

General | Events Selection

Trace name: SQLProfiler

Trace provider name: LEGACIES\SQLEXPRESS

Trace provider type: Microsoft SQL Server "2022" version: 16.0.1000

Use the template: Standard (default)

☐ Save to file:   
Set maximum file size (MB): 5   
☒ Enable file rollover   
☐ Server processes trace data

☐ Save to table:   
☐ Set maximum rows (in thousands): 1

☐ Enable trace stop time: 4/22/2025 2:55:28 PM   
☒ Set trace duration (in minutes): 60

Run Cancel Help

Trace Properties

General | Events Selection

Review selected events and event columns to trace. To see a complete list, select the "Show all events" and "Show all columns" options.

Events	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProc
Deprecation									
Errors and Warnings									
<input checked="" type="checkbox"/> Attention		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Background Job Error									
<input checked="" type="checkbox"/> Bitmap Warning		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Blocked process report	<input checked="" type="checkbox"/>							<input checked="" type="checkbox"/>	
<input checked="" type="checkbox"/> CPU threshold exceeded					<input checked="" type="checkbox"/>				
<input checked="" type="checkbox"/> Database Suspect Data Page									
<input checked="" type="checkbox"/> ErrorLog	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> EventLog	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Exception	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Exchange Spill Event		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>					<input checked="" type="checkbox"/>

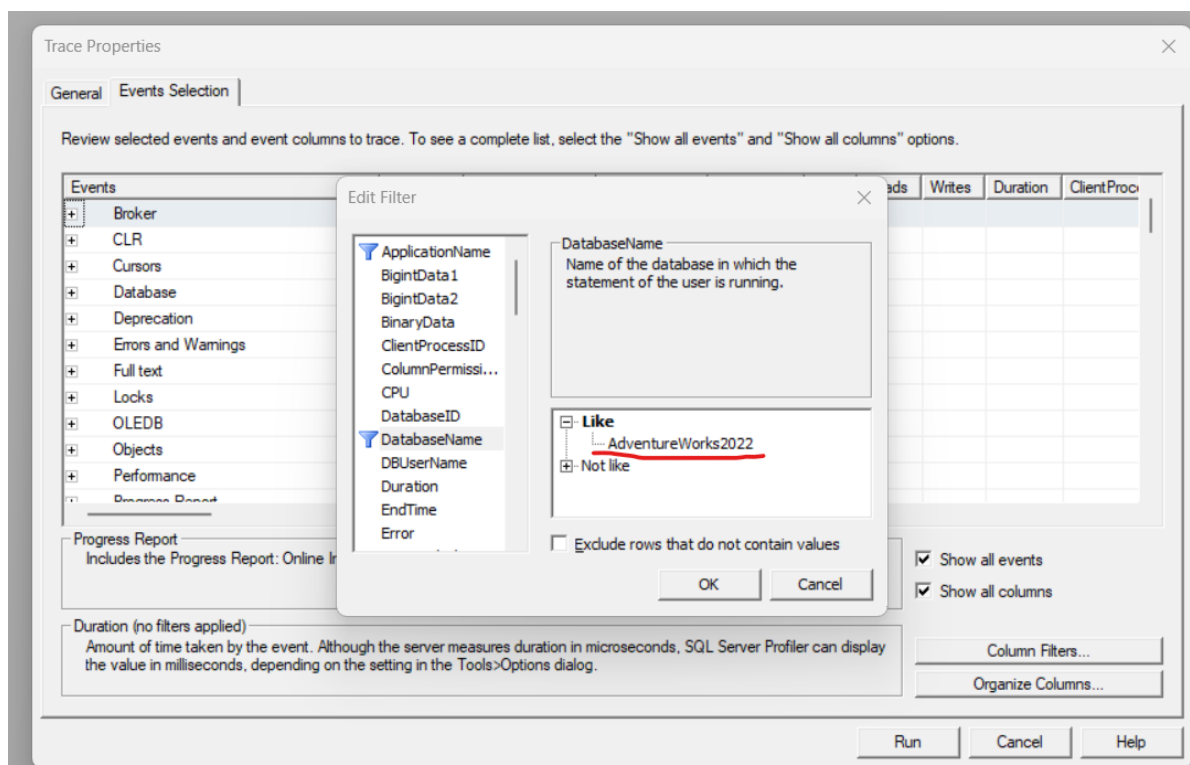
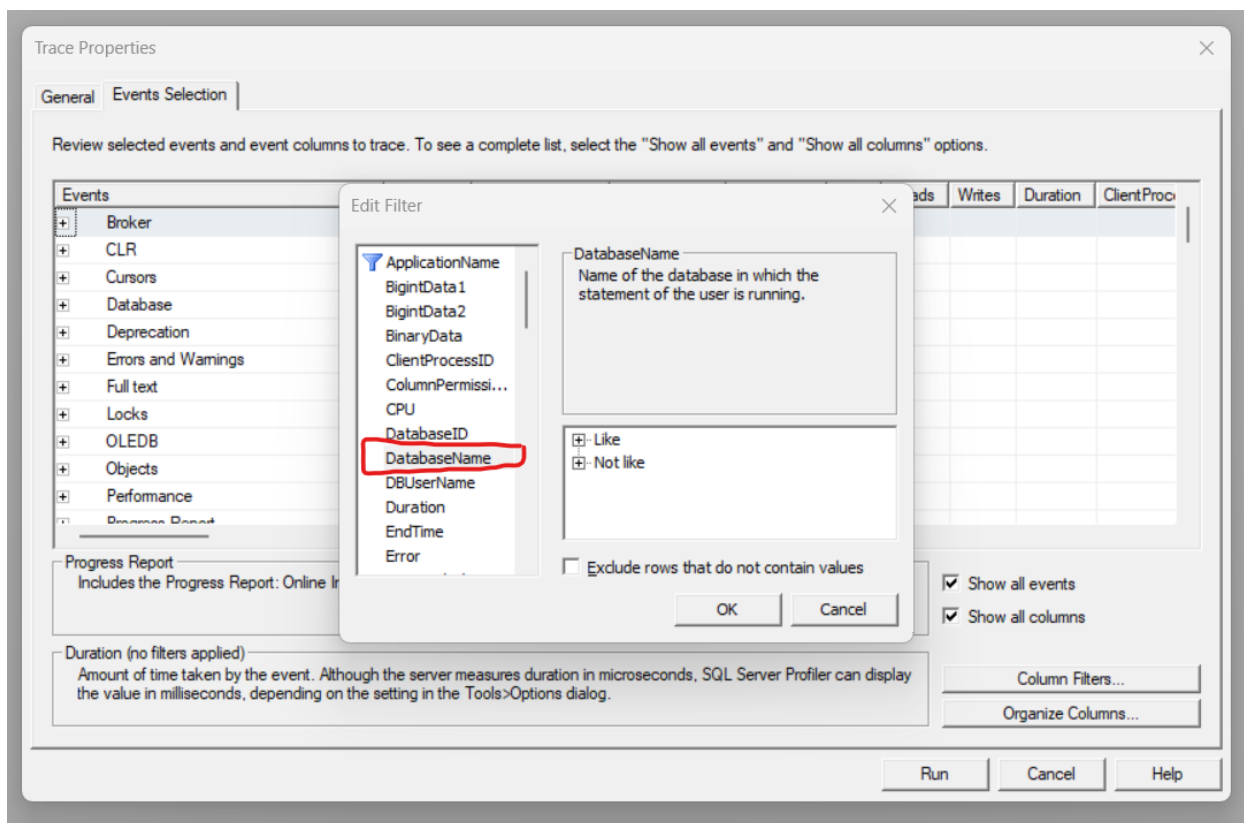
Exchange Spill Event  
The Exchange Spill event class indicates that communication buffers in a parallel query plan have been temporarily written to the tempdb database. This occurs rarely and only when a query plan has multiple range scans.

☒ Show all events  
☒ Show all columns

Column Filters...  
Organize Columns...

CPU (no filters applied)  
Amount of CPU time (in milliseconds) used by the event.

Run Cancel Help





SQL:BatchStarting	SELECT * FROM HumanResources.Department...	Microsoft SQ...	jedim	Legaci...					18480	73	2025-04-23 13:32:03...
SQL:BatchCompleted	SELECT * FROM HumanResources.Department...	Microsoft SQ...	jedim	Legaci...	0	67	0	6	18480	73	2025-04-23 13:32:03...
SQL:BatchStarting	SELECT * FROM HumanResources.Department...	Microsoft SQ...	jedim	Legaci...					18480	73	2025-04-23 13:32:43...

SELECT \* FROM HumanResources.Department

SQLQuery1.sql - LE...GACIES\jedim (73))\*

SELECT \* FROM HumanResources.Department WHERE DepartmentID == 300

100 %

Messages

Msg 102, Level 15, State 1, Line 1  
Incorrect syntax near '='.

Completion time: 2025-04-23T13:32:51.7250204+03:00

Exception	Incorrect syntax near '='.	Microsoft SQ...	jedim	Legaci...					18480	73	2025-04-23 13:32:51...
User Error Message	Incorrect syntax near '='.	Microsoft SQ...	jedim	Legaci...					18480	73	2025-04-23 13:32:51...
SQL:BatchStarting	SELECT * FROM HumanResources.Department...	Microsoft SQ...	jedim	Legaci...					18480	73	2025-04-23 13:32:51...

Incorrect syntax near '='.

## b. İndeks Yönetimi:

```
SQLQuery7.sql - LE...GACIES\jedim (81))* X SQLQuery9.sql - LE...GACIES\jedim (52))* SQLQuery8.sql - LE...GACIES\jedim
USE AdventureWorks2022
SELECT
    OBJECT_NAME(i.object_id) AS TableName,
    i.name AS IndexName,
    i.type_desc,
    ISNULL(s.user_seeks, 0) AS user_seeks,
    ISNULL(s.user_scans, 0) AS user_scans,
    ISNULL(s.user_lookups, 0) AS user_lookups,
    ISNULL(s.user_updates, 0) AS user_updates,
    s.last_user_seek,
    s.last_user_scan,
    s.last_user_lookup
FROM sys.indexes i
LEFT JOIN sys.dm_db_index_usage_stats s
    ON i.object_id = s.object_id AND i.index_id = s.index_id
WHERE OBJECTPROPERTY(i.object_id, 'IsUserTable') = 1
    AND i.name IS NOT NULL
ORDER BY (ISNULL(s.user_seeks, 0) + ISNULL(s.user_scans, 0) + ISNULL(s.user_lookups, 0)) ASC;
```

100 %

Messages

SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 0 ms.

Completion time: 2025-04-24T02:52:19.9895584+03:00

Sistemde kullanılmayan ve gereksiz olan indeksleri bulmayı sağlayan sql kodudur.

```
SQLQuery7.sql - LE...GACIES\jedim (62))* X SQLQuery9.sql - LE...GACIES\jedim (52))* SQLQuery8.sql - LE...GACIES\jedim (62))*
USE AdventureWorks2022
SELECT
    OBJECT_NAME(i.object_id) AS TableName,
    i.name AS IndexName,
    i.type_desc,
    ISNULL(s.user_seeks, 0) AS user_seeks,
    ISNULL(s.user_scans, 0) AS user_scans,
    ISNULL(s.user_lookups, 0) AS user_lookups,
    ISNULL(s.user_updates, 0) AS user_updates,
    s.last_user_seek,
    s.last_user_scan,
    s.last_user_lookup
FROM sys.indexes i
LEFT JOIN sys.dm_db_index_usage_stats s
    ON i.object_id = s.object_id AND i.index_id = s.index_id
WHERE OBJECTPROPERTY(i.object_id, 'IsUserTable') = 1
    AND i.name IS NOT NULL
ORDER BY (ISNULL(s.user_seeks, 0) + ISNULL(s.user_scans, 0) + ISNULL(s.user_lookups, 0)) ASC;
```

100 %

Results Messages

	TableName	IndexName	type_desc	user_seeks	user_scans	user_lookups	user_updates
34	DatabaseLog	PK_DatabaseLog_DatabaseLogID	NONCLUSTERED	0	0	0	0
35	ProductInventory	PK_ProductInventory_ProductID_LocationID	CLUSTERED	0	0	0	0
36	SpecialOffer	PK_SpecialOffer_SpecialOfferID	CLUSTERED	0	0	0	0
37	SpecialOffer	AK_SpecialOffer_rowguid	NONCLUSTERED	0	0	0	0
38	ErrorLog	PK_ErrorLog_ErrorLogID	CLUSTERED	0	0	0	0
39	ProductListPriceHistory	PK_ProductListPriceHistory_ProductID_StartDate	CLUSTERED	0	0	0	0
40	Address	PK_Address_AddressID	CLUSTERED	0	0	0	0
41	Address	AK_Address_rowguid	NONCLUSTERED	0	0	0	0
42	Address	IX_Address_AddressLine1_AddressLine2_City_StateProvinceID	NONCLUSTERED	0	0	0	0
43	Address	IX_Address_StateProvinceID	NONCLUSTERED	0	0	0	0
44	SpecialOfferProduct	PK_SpecialOfferProduct_SpecialOfferID_ProductID	CLUSTERED	0	0	0	0
45	SpecialOfferProduct	AK_SpecialOfferProduct_rowguid	NONCLUSTERED	0	0	0	0
46	SpecialOfferProduct	IX_SpecialOfferProduct_ProductID	NONCLUSTERED	0	0	0	0



İndeksleri kaldırırken PK ve AK olmayanları seçmeliyiz çünkü PK ve AK database'lerde constraint olarak tutulmaktadır. Bu tarz indeksleri silerken permission hatası alırsın. Bu yüzden ismi IX olanların kaldırılması mümkündür. Aşağıdaki iki resimde, kaldırmak için IX olarak seçilenler ve indeks kaldırma kodları yer almaktadır.

```
DROP INDEX IX_ShoppingCartItem_ShoppingCartID_ProductID ON Sales.ShoppingCartItem;
```

```
DROP INDEX IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode ON Person.Address;
```

Bu iki tane indeks kaldırma komutunu çalıştırmadan önce iki tane base query belirleyip. Bu iki base query'i indeks kaldırma öncesi ve sonrası olmak üzere her biri için 2 defa çalıştırıp Sql profiler kısmından duration süresindeki düşüşü (hız artışı) gözlemleyeceğiz.

SQL:BatchStarting	USE Adventureworks2022	SELECT * F...	Microsoft SQ...	jedim	Legaci...				
SQL:BatchCompleted	USE Adventureworks2022	SELECT * F...	Microsoft SQ...	jedim	Legaci...	0	210	0	3
SQL:BatchStarting	DROP INDEX IX_ShoppingCartItem_Shopp...		Microsoft SQ...	jedim	Legaci...				
SQL:BatchCompleted	DROP INDEX IX_ShoppingCartItem_Shopp...		Microsoft SQ...	jedim	Legaci...	0	114	15	4
SQL:BatchStarting	SELECT * FROM Sales.ShoppingCartItem		Microsoft SQ...	jedim	Legaci...				
SQL:BatchCompleted	SELECT * FROM Sales.ShoppingCartItem		Microsoft SQ...	jedim	Legaci...	0	24	0	2

ELECT \* FROM Sales.ShoppingCartItem

Sorgunun 3 tane sonucu olduğu için bu kadar az sample'ın bulunduğu bir query yerine AddressLine için query yazıp fazlaca büyük sample sayısına sahip query'lerdeki performans artışını gözlemleyeceğiz.

SQL:BatchCompleted	SELECT * FROM Person.Address	Microsoft SQ...	jedim	Legaci...	31	666	0	273
SQL:BatchStarting	DROP INDEX IX_Address_AddressLine1...	Microsoft SQ...	jedim	Legaci...				
SQL:BatchCompleted	DROP INDEX IX_Address_AddressLine1...	Microsoft SQ...	jedim	Legaci...	0	253	10	4
SQL:BatchStarting	SELECT * FROM Person.Address	Microsoft SQ...	jedim	Legaci...				
SQL:BatchCompleted	SELECT * FROM Person.Address	Microsoft SQ...	jedim	Legaci...	16	364	0	146

SELECT \* FROM Person.Address

Yaklaşık 2 kat kadar hız artışı vardır. (273 → 146)

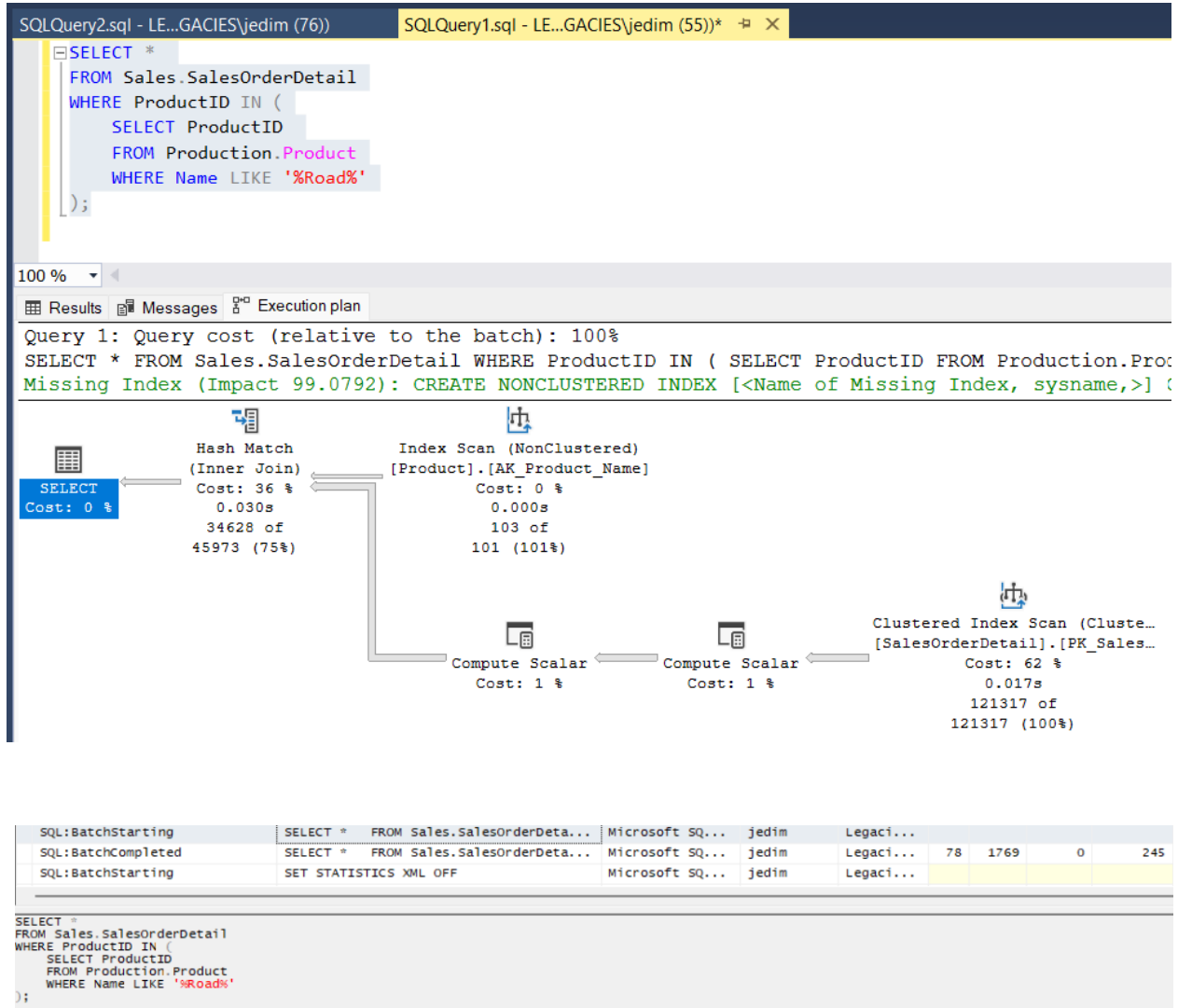
### c. Sorgu İyileştirme:

Sql'de bir query'de kullanılabilir her keyword'ün database'de oluşturduğu sistem yükleri vardır.

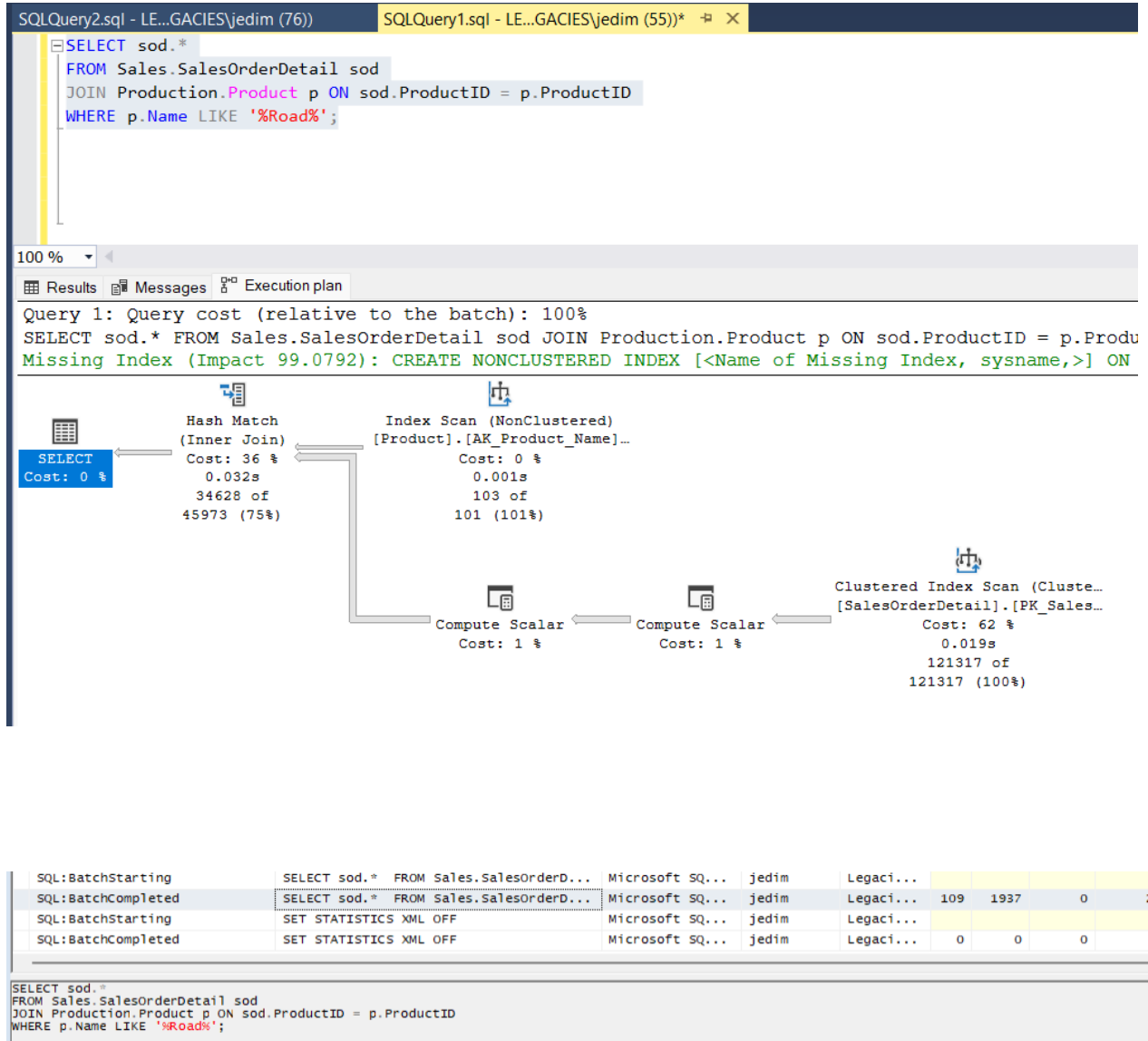
Keyword 1	Keyword 2	Hangisi daha hızlı?
Where	Having	Where
Select *	Select col1, col2 ...	Select col1, col2, ...
In	Join	Join

Aşağıdaki tüm query'lerin sonuçları birebir aynıdır!

## 1. Sorgu (in kullanımı)



## 2. Sorgu (Join kullanımı)



### 3. Sorgu (Join + Select col1, col2, ...)

SQLQuery2.sql - LE...GACIES\jedim (76)    SQLQuery1.sql - LE...GACIES\jedim (55))\*

```
SELECT sod.SalesOrderID, sod.SalesOrderDetailID, sod.CarrierTrackingNumber
FROM Sales.SalesOrderDetail sod
JOIN Production.Product p ON sod.ProductID = p.ProductID
WHERE p.Name LIKE '%Road%';
```

100 %

Results    Messages    Execution plan

Query 1: Query cost (relative to the batch): 100%

SELECT sod.SalesOrderID, sod.SalesOrderDetailID, sod.CarrierTrackingN

Missing Index (Impact 99.2437): CREATE NONCLUSTERED INDEX [<Name of M

```
graph LR
    A[SELECT  
Cost: 0 %] --> B[Hash Match  
(Inner Join)  
Cost: 37 %  
0.031s  
34628 of  
45973 (75%)]
    B --> C[Index Scan (NonClustered)  
[Product].[AK_Product_Name]...  
Cost: 0 %  
0.000s  
103 of  
101 (101%)]
    B --> D[Clustered Index Scan (Cluste...  
[SalesOrderDetail].[PK_Sales...  
Cost: 63 %  
0.018s  
121317 of  
121317 (100%)]
```

SQL:BatchStarting	SELECT sod.SalesOrderID, sod.SalesOr...	Microsoft SQ...	jedim	Legaci...				
SQL:BatchCompleted	SELECT sod.SalesOrderID, sod.SalesOr...	Microsoft SQ...	jedim	Legaci...	32	1937	0	132
SQL:BatchStarting	SET STATISTICS XML OFF	Microsoft SQ...	jedim	Legaci...				
SQL:BatchCompleted	SET STATISTICS XML OFF	Microsoft SQ...	jedim	Legaci...	0	0	0	0

```
SELECT sod.SalesOrderID, sod.SalesOrderDetailID, sod.CarrierTrackingNumber
FROM Sales.SalesOrderDetail sod
JOIN Production.Product p ON sod.ProductID = p.ProductID
WHERE p.Name LIKE '%Road%';
```

#### d. Veri Yöneticisi Roller:

```
SQLQuery2.sql - LE...GACIES\jedim (76))  SQLQuery1.sql - LE...GACIES\jedim (55))* + X
CREATE LOGIN report_user_login WITH PASSWORD = 'Report123!';
CREATE LOGIN editor_user_login WITH PASSWORD = 'Edit123!';
CREATE LOGIN admin_user_login WITH PASSWORD = 'Admin123!';

100 %

Messages

SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 2 ms.
SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 0 ms.

Completion time: 2025-04-22T15:32:58.0878185+03:00
```

Sql server authentication için login oluşturulması lazım.

```
USE AdventureWorks2022;
CREATE USER report_user FOR LOGIN report_user_login;
CREATE USER editor_user FOR LOGIN editor_user_login;
CREATE USER admin_user FOR LOGIN admin_user_login;

100 %

Messages

SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 2 ms.

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time:
  CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
  CPU time = 0 ms, elapsed time = 0 ms.

Completion time: 2025-04-22T15:33:30.8323288+03:00
```

Sonrasında kullanıcılar oluşturulmalı ve login'e bağlanmalı.

```
CREATE ROLE report_role;  
CREATE ROLE editor_role;  
CREATE ROLE admin_role;
```

100 %

#### Messages

Commands completed successfully.

Veritabanına bu kullanıcılara yetkilendirme verebilmek için roller oluşturulmalı.

```
GRANT SELECT ON SCHEMA::Sales TO report_role;  
GRANT SELECT, UPDATE ON SCHEMA::Sales TO editor_role;  
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::Sales TO admin_role;
```

100 %

#### Messages

```
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.  
  
SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 0 ms.  
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.  
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.  
  
SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 0 ms.  
  
Completion time: 2025-04-22T15:37:05.9096362+03:00
```

Rollere yetkilendirme verilmeli.

```
EXEC sp_addrolemember 'report_role', 'report_user';  
EXEC sp_addrolemember 'editor_role', 'editor_user';  
EXEC sp_addrolemember 'admin_role', 'admin_user';
```

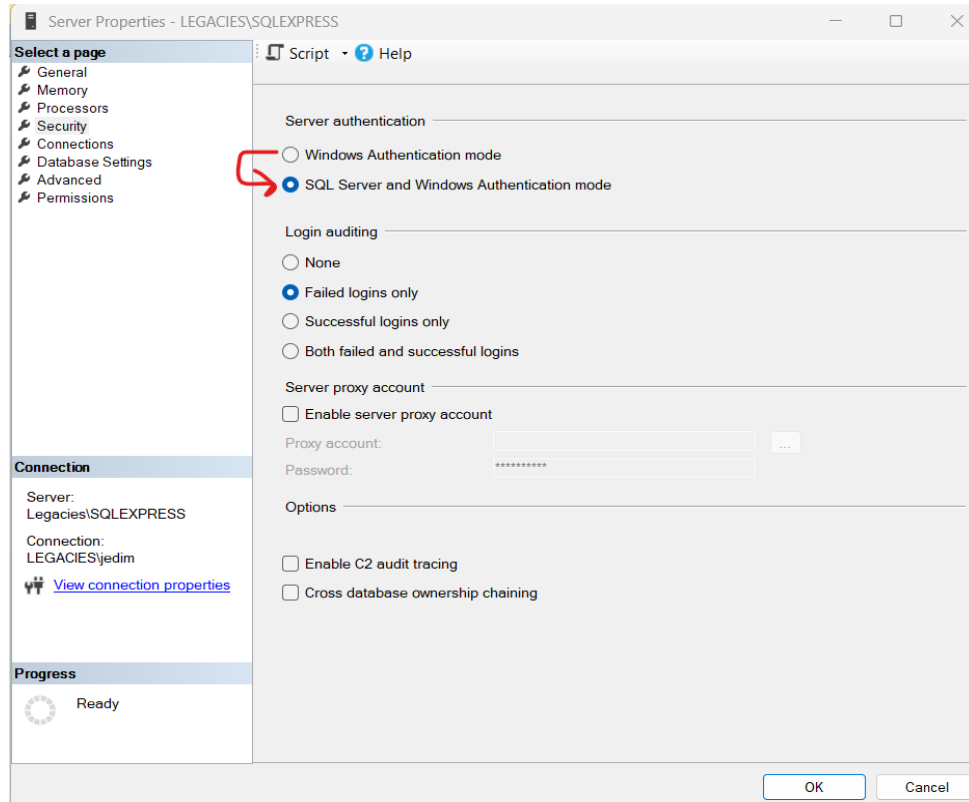
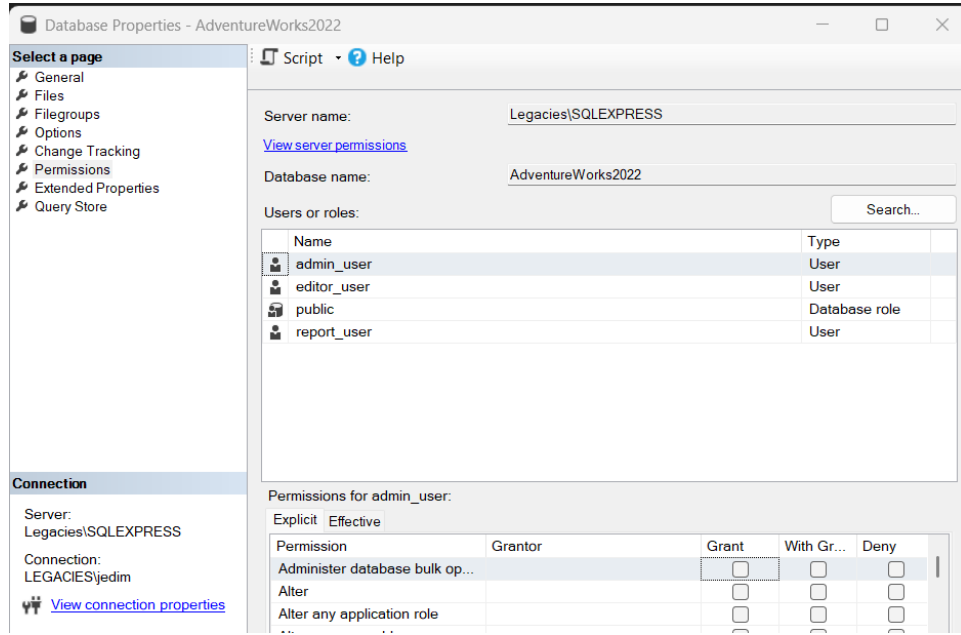
100 %

#### Messages

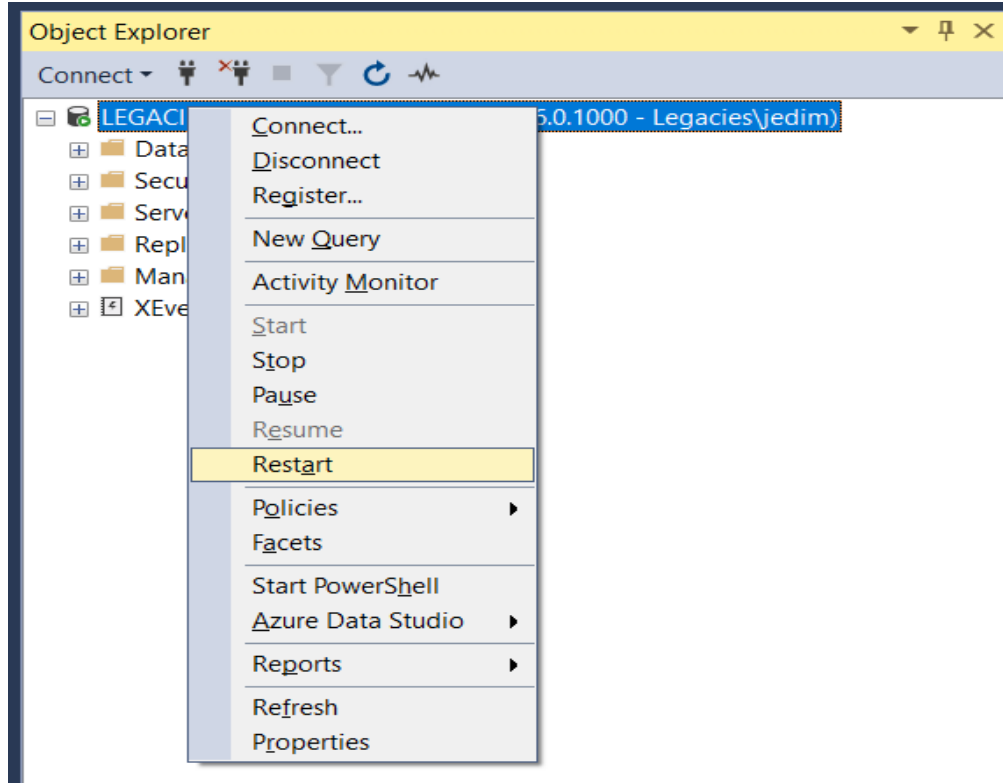
```
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.  
  
SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 0 ms.  
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 2 ms.  
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 2 ms.
```

Roller veritabanına sp\_addrolemember komutu ile assign edilmeli.

Hata almamak için database ayarlarından sql server authentication özelliğinin aktif edilmesi lazım.

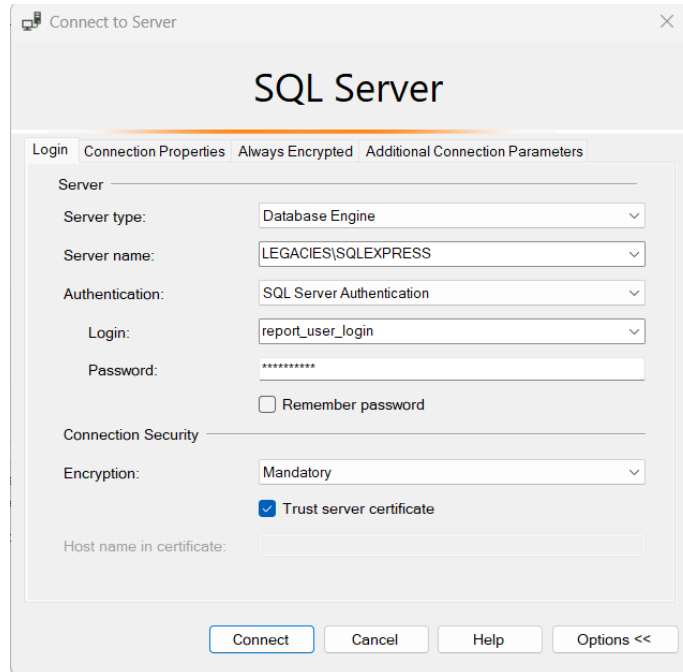


Aktif olduğundan emin olmak için server'ın aktif edilmesi lazım.



Eklenen kullanıcılarla giriş yapıp yetkilendirmelerinin doğru çalışıp çalışmadığını test edelim.

### Report Kullanıcısı için:





SQLQuery2.sql - LEG...t\_user\_login (107))\* X SQLQuery1.sql - LE...GACIES\jedim (60))\*

```
SELECT TOP 10 * FROM Sales.SalesOrderHeader; -- İzin verilen
```

100 %

Results Messages

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate
1	43659	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
2	43660	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
3	43661	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
4	43662	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
5	43663	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
6	43664	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
7	43665	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
8	43666	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
9	43667	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000
10	43668	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000

```
UPDATE Sales.SalesOrderHeader SET Comment = 'Test' WHERE SalesOrderID = 10; -- İzin verilmeyen
```

100 %

Messages

Msg 229, Level 14, State 5, Line 3  
The UPDATE permission was denied on the object 'SalesOrderHeader', database 'AdventureWorks2022', schema 'Sales'.

Completion time: 2025-04-22T16:04:26.2790325+03:00

## Editor Kullanıcısı için:

Connect to Server

### SQL Server

Login Connection Properties Always Encrypted Additional Connection Parameters

Server

Server type: Database Engine

Server name: LEGACIES\SQLEXPRESS

Authentication: SQL Server Authentication

Login: editor\_user\_login

Password: \*\*\*\*\*

☐ Remember password

Connection Security

Encryption: Mandatory

☒ Trust server certificate

Host name in certificate:

Connect Cancel Help Options <<

SQLQuery3.sql - LEG...or\_user\_login (67))\*

```
SELECT * FROM Sales.SalesOrderHeader;
```

100 %

Results Messages

	SalesOrderID	RevisionNumber	OrderDate	De
1	43659	8	2011-05-31 00:00:00.000	20
2	43660	8	2011-05-31 00:00:00.000	20
3	43661	8	2011-05-31 00:00:00.000	20
4	43662	8	2011-05-31 00:00:00.000	20
5	43663	8	2011-05-31 00:00:00.000	20
6	43664	8	2011-05-31 00:00:00.000	20
7	43665	8	2011-05-31 00:00:00.000	20
8	43666	8	2011-05-31 00:00:00.000	20
9	43667	8	2011-05-31 00:00:00.000	20
10	43668	8	2011-05-31 00:00:00.000	20
11	43669	8	2011-05-31 00:00:00.000	20
12	43670	8	2011-05-31 00:00:00.000	20

```
SELECT SalesOrderID, Comment FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43659;
UPDATE Sales.SalesOrderHeader SET Comment = 'Updated' WHERE SalesOrderID = 43659;
SELECT SalesOrderID, Comment FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43659;
```

100 %

Results Messages

	SalesOrderID	Comment
1	43659	NULL

	SalesOrderID	Comment
1	43659	Updated

```
DELETE FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43659;
```

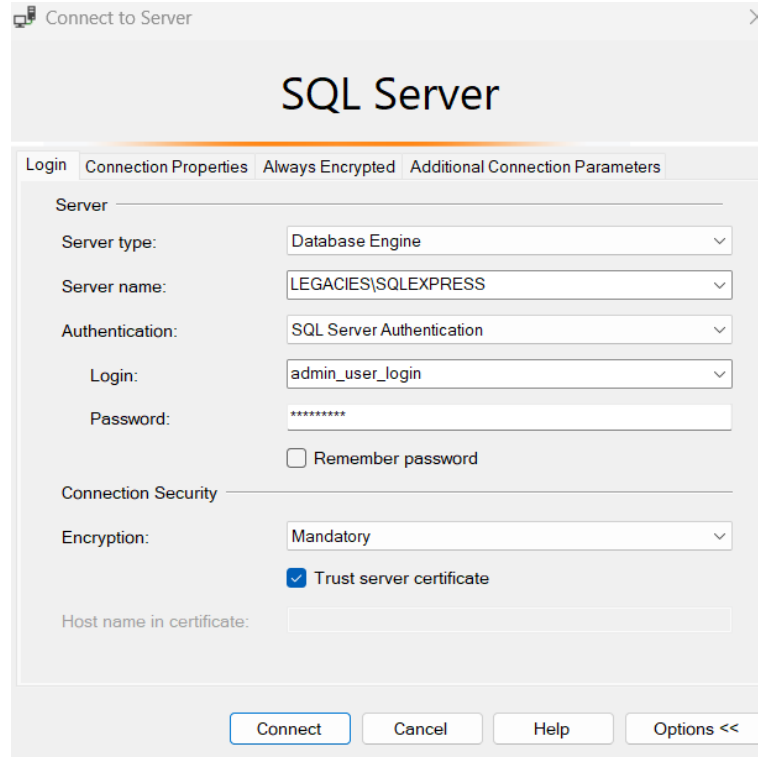
100 %

Messages

Msg 229, Level 14, State 5, Line 7  
The DELETE permission was denied on the object 'SalesOrderHeader', database 'AdventureWorks2022', schema 'Sales'.

Completion time: 2025-04-22T16:12:23.6185134+03:00

## Admin Kullanıcısı için:



The image shows the 'Connect to Server' dialog box in SQL Server Enterprise Manager. The 'Server' tab is selected. The 'Server type' is set to 'Database Engine'. The 'Server name' is 'LEGACIES\SQLEXPRESS'. The 'Authentication' is 'SQL Server Authentication'. The 'Login' is 'admin\_user\_login' and the 'Password' is masked with asterisks. The 'Remember password' checkbox is unchecked. Under 'Connection Security', the 'Encryption' is set to 'Mandatory' and the 'Trust server certificate' checkbox is checked. The 'Host name in certificate' field is empty. At the bottom, there are buttons for 'Connect', 'Cancel', 'Help', and 'Options <<'. The title bar of the dialog says 'Connect to Server'.

SQL Server

Login Connection Properties Always Encrypted Additional Connection Parameters

Server

Server type: Database Engine

Server name: LEGACIES\SQLEXPRESS

Authentication: SQL Server Authentication

Login: admin\_user\_login

Password: \*\*\*\*\*

☐ Remember password

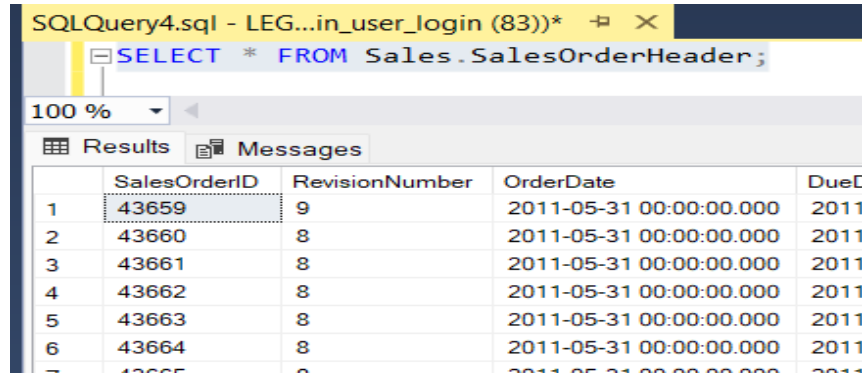
Connection Security

Encryption: Mandatory

☒ Trust server certificate

Host name in certificate:

Connect Cancel Help Options <<



The image shows a SQL Server Query Results window. The title bar says 'SQLQuery4.sql - LEG...in\_user\_login (83)\*'. The query editor shows the query: 'SELECT \* FROM Sales.SalesOrderHeader;'. The 'Results' tab is selected, showing a table with 5 columns: 'SalesOrderID', 'RevisionNumber', 'OrderDate', and 'DueDate'. The table contains 6 rows of data. The first row is highlighted. The zoom level is set to 100%.

SQLQuery4.sql - LEG...in\_user\_login (83)\*

SELECT \* FROM Sales.SalesOrderHeader;

100 %

Results Messages

	SalesOrderID	RevisionNumber	OrderDate	DueDate
1	43659	9	2011-05-31 00:00:00.000	2011
2	43660	8	2011-05-31 00:00:00.000	2011
3	43661	8	2011-05-31 00:00:00.000	2011
4	43662	8	2011-05-31 00:00:00.000	2011
5	43663	8	2011-05-31 00:00:00.000	2011
6	43664	8	2011-05-31 00:00:00.000	2011

SQLQuery4.sql - LEG...in\_user\_login (83))\*

```

SELECT * FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43659;
DELETE FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43659;
SELECT * FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43659;

```

100 %

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate
1	43659	9	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000

Results Messages

SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrd
--------------	----------------	-----------	---------	----------	--------	-----------------	----------

```

SELECT SalesOrderID, Comment FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43660;
UPDATE Sales.SalesOrderHeader SET Comment = 'Admin OK' WHERE SalesOrderID = 43660;
SELECT SalesOrderID, Comment FROM Sales.SalesOrderHeader WHERE SalesOrderID = 43660;

```

100 %

	SalesOrderID	Comment
1	43660	NULL

Results Messages

	SalesOrderID	Comment
1	43660	Admin OK

```

SELECT * FROM Sales.SalesOrderHeader WHERE RevisionNumber = 1;
INSERT INTO [AdventureWorks2022].[Sales].[SalesOrderHeader] (RevisionNumber, OrderDate, DueDate, ShipDate, Status, OnlineOrderFlag,
PurchaseOrderNumber, AccountNumber, CustomerID, SalesPersonID, TerritoryID, BillToAddressID, ShipToAddressID,
ShipMethodID, CreditCardID, CreditCardApprovalCode, CurrencyRateID, SubTotal, TaxAmt, Freight, Comment, rowguid,
ModifiedDate)
VALUES (
1, GETDATE(), DATEADD(DAY, 5, GETDATE()), DATEADD(DAY, 7, GETDATE()), 5, 1, N'PO999999', N'AC1234567', 11000, 279, 3,
1, 1, 4, 10, N'APPROVED123', 1, 100.00, 10.00, 5.00, N'Test Order Inserted', NEWID(), GETDATE()
);
SELECT * FROM Sales.SalesOrderHeader WHERE RevisionNumber = 1;

```

100 %

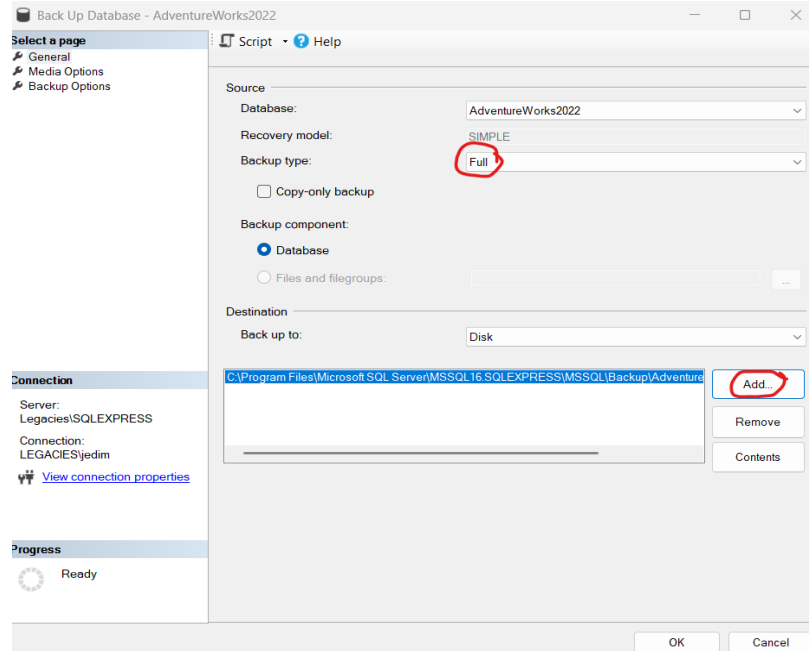
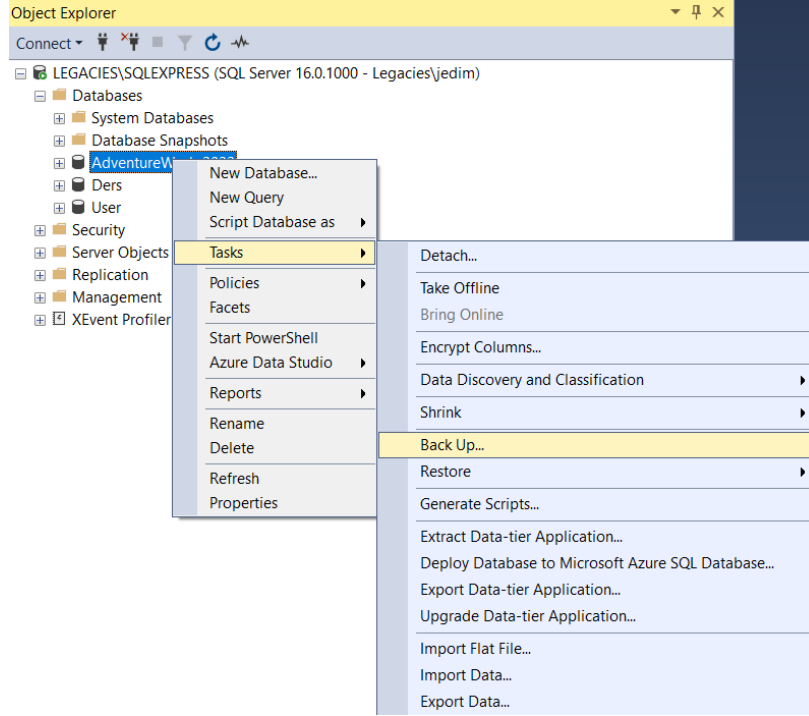
	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	PurchaseOrderNumber	AccountNumber	CustomerID	SalesPersonID	TerritoryID	BillToAddressID
1	75125	1	2025-04-22 16:28:21.360	2025-04-27 16:28:21.360	2025-04-29 16:28:21.360	5	1	SO75125	PO99999	AC1234567	11000			

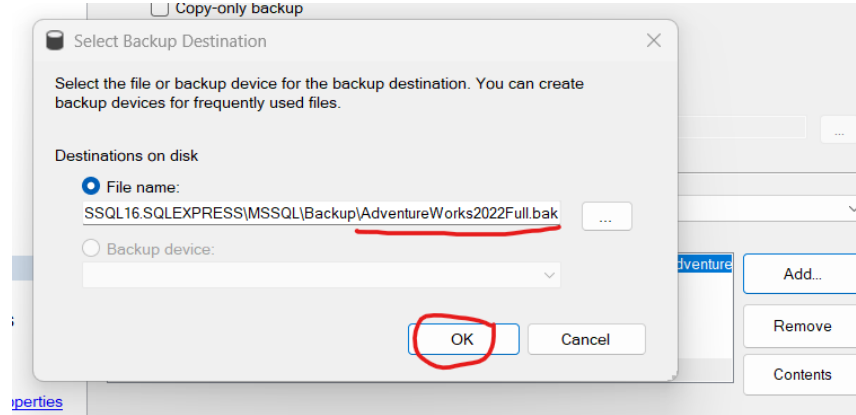
Results Messages

## 2. Veritabanı Yedekleme ve Felaketten Kurtarma Planı

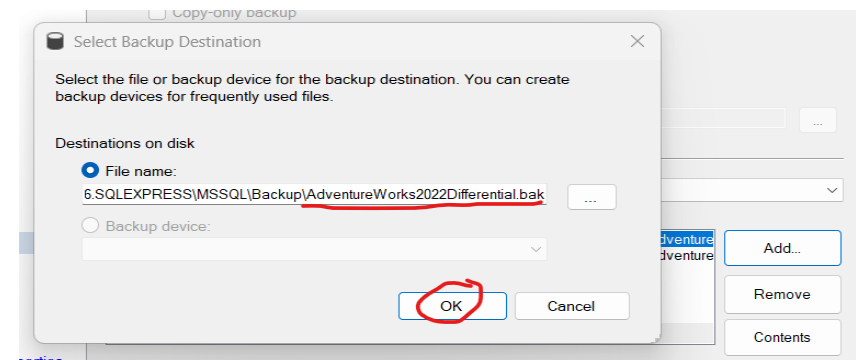
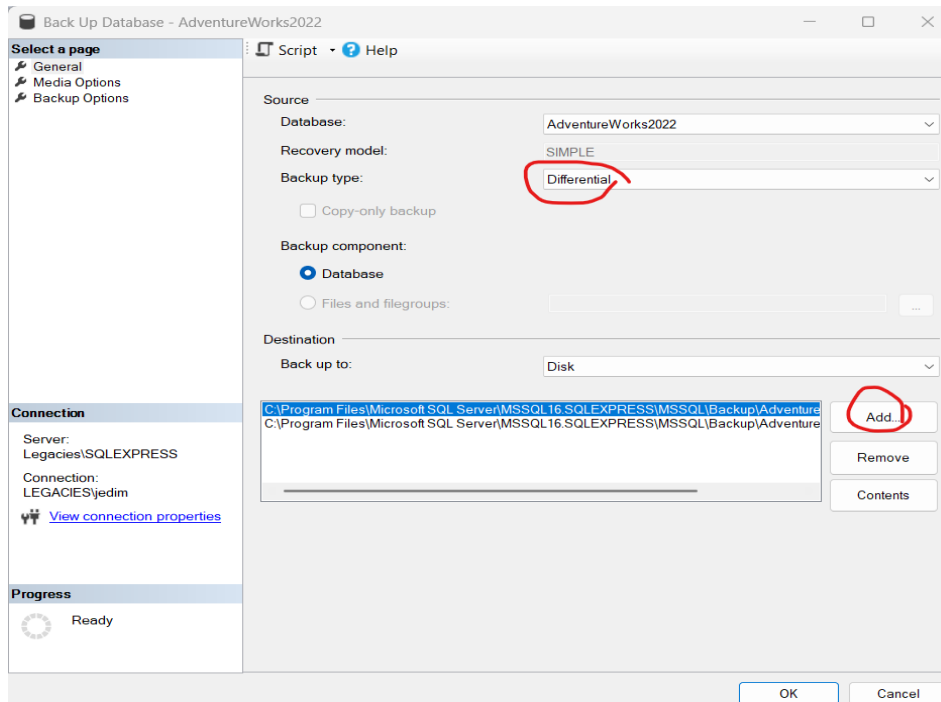
### a. Tam, Artık ve Fark Yedeklemeleri:

#### Tam Yedekleme:

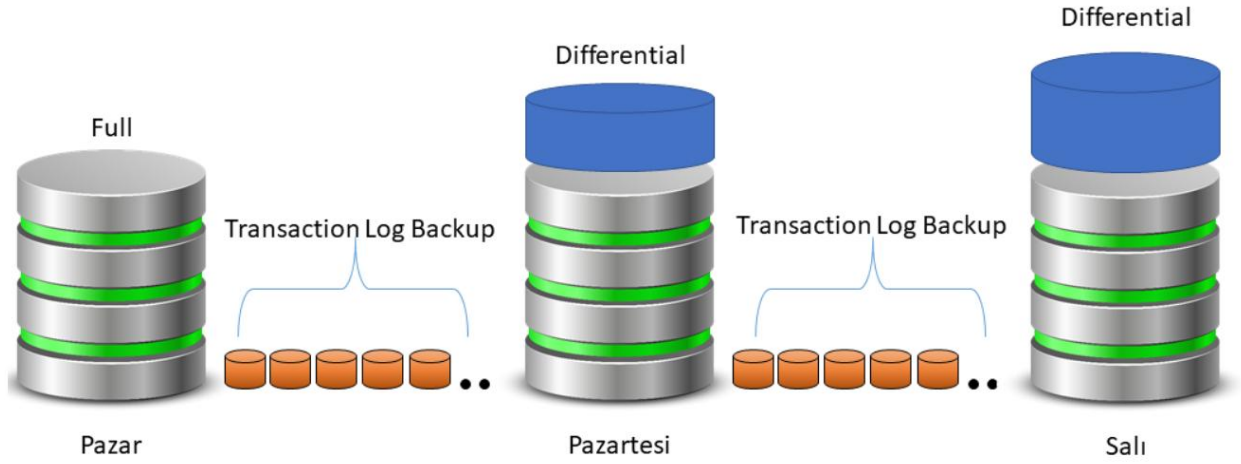




## Fark Yedekleme:

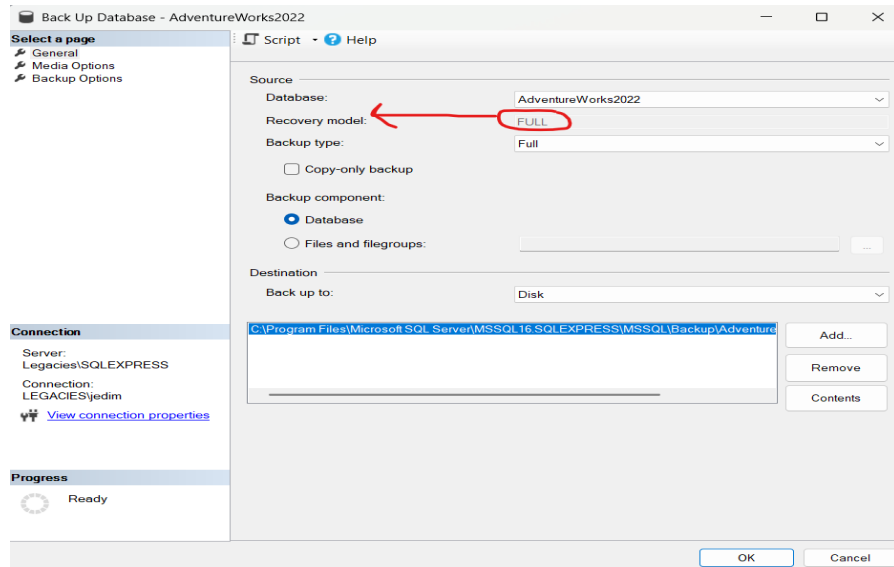


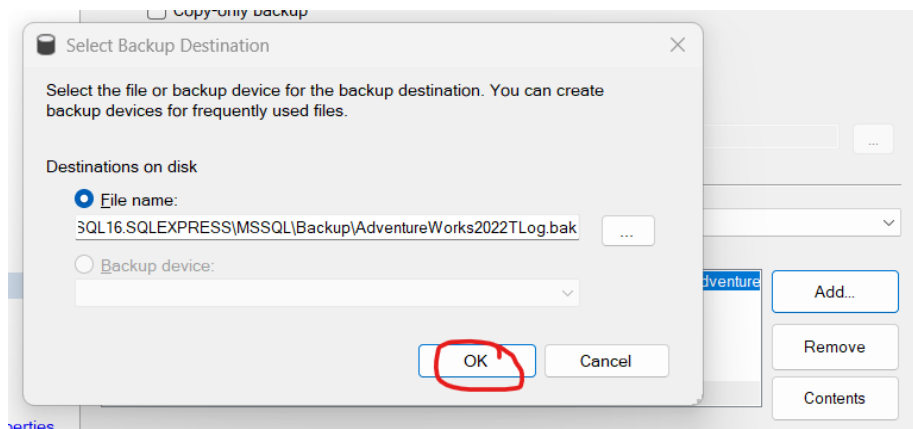
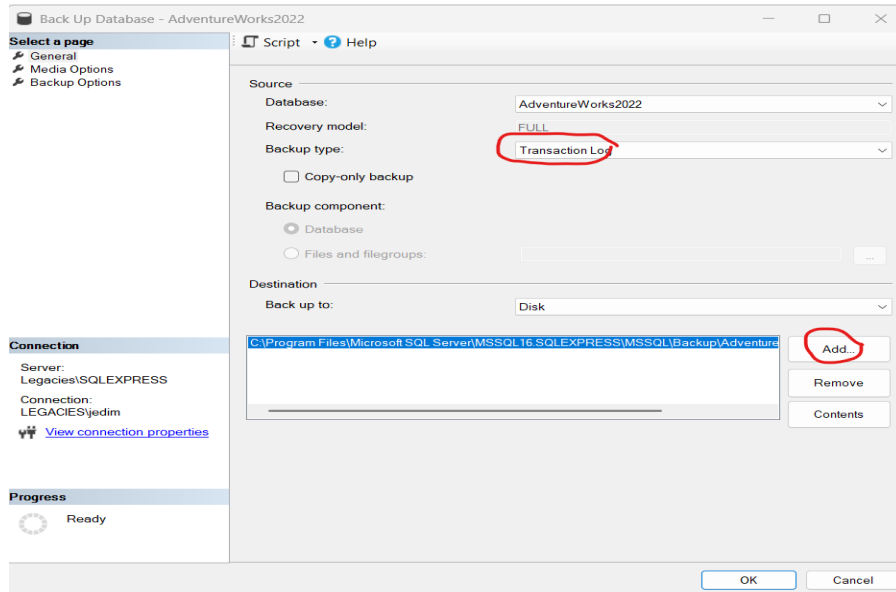
## Artık Yedekleme:



Artık yedekleme yapabilmek için öncesinde full yedeklemenin yapılmış olması lazım ve recovery modun FULL olması lazım çünkü artık yedeklemede

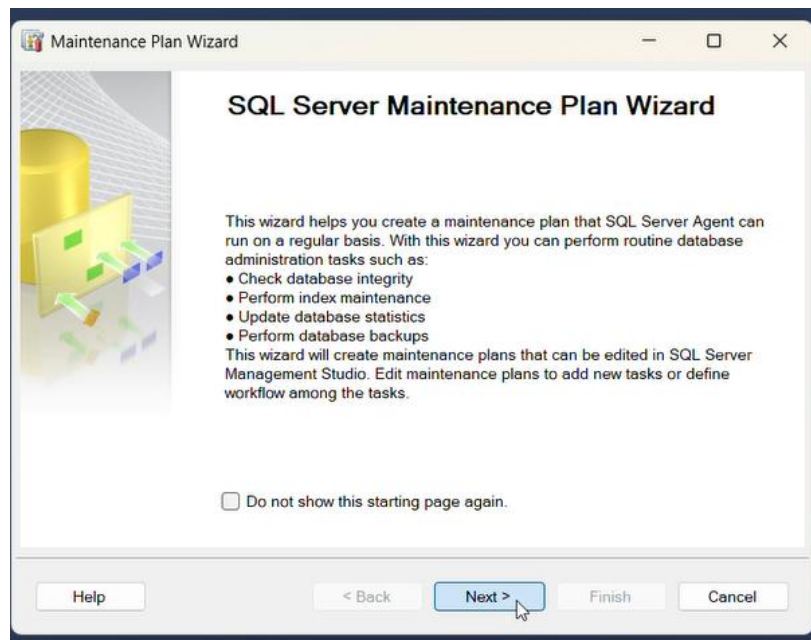
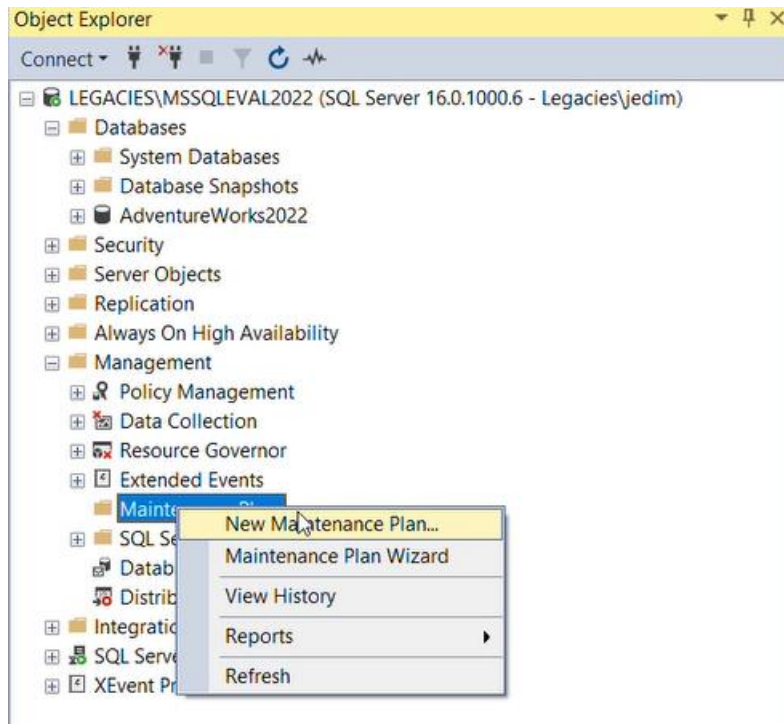
```
SQLQuery1.sql - LE...GACIES\jedim (53))*  
USE [master]  
ALTER DATABASE AdventureWorks2022 SET RECOVERY FULL;  
100 %  
Messages  
Commands completed successfully.  
Completion time: 2025-04-22T21:59:45.8727157+03:00
```







## b. Zamanlayıcılarla Yedeklemeleri:



Maintenance Plan Wizard

Select Plan Properties  
How do you want to schedule your maintenance tasks?

Name: MaintenancePlan

Description:

Run as: SQL Server Agent service account

☐ Separate schedules for each task  
☒ Single schedule for the entire plan or no schedule

Schedule: Not scheduled (On Demand) Change...

Help < Back Next > Finish Cancel

New Job Schedule

Name: MaintenancePlan

Schedule type: Recurring ☒ Enabled

One-time occurrence  
Date: 4/23/2025 Time: 5:39:40 PM

Frequency  
Occurs: Daily  
Recurs every: 1 day(s)

Daily frequency  
☒ Occurs once at: 12:00:00 AM  
☐ Occurs every: 1 hour(s)

Starting at: 12:00:00 AM  
Ending at: 11:59:59 PM

Duration  
Start date: 4/23/2025 ☐ End date: 4/23/2025  
☒ No end date

Summary  
Description: Occurs every day at 12:00:00 AM. Schedule will be used starting on 4/23/2025.

OK Cancel Help

Maintenance Plan Wizard

### Select Maintenance Tasks

Which tasks should this plan perform?

Select one or more maintenance tasks:

- ☐ Check Database Integrity
- ☐ Shrink Database
- ☐ Reorganize Index
- ☐ Rebuild Index
- ☐ Update Statistics
- ☐ Clean Up History
- ☐ Execute SQL Server Agent Job
- ☒ Back Up Database (Full)
- ☐ Back Up Database (Differential)
- ☐ Back Up Database (Transaction Log)
- ☐ Maintenance Cleanup Task

The Back Up Database (Full) task allows you to specify the source databases, destination files or tapes, and overwrite options for a full backup.

Help < Back Next > Finish Cancel

Maintenance Plan Wizard

### Select Maintenance Task Order

In which order should these tasks be performed?

Select the order for the tasks to execute:

- Back Up Database (Full)

Move Up... Move Down...

The Back Up Database (Full) task allows you to specify the source databases, destination files or tapes, and overwrite options for a full backup.

Help < Back Next > Finish Cancel

Maintenance Plan Wizard

### Define Back Up Database (Full) Task

Configure the maintenance task.

General Destination Options

Backup type: Full

Database(s): <Select one or more>

Backup component

☒ Database

☐ Files and filegroups:

Back up to:

Not scheduled (On Demand)

Help

☐ All databases

☐ System databases

☐ All user databases (excluding master, model, msdb, tempdb)

☒ These databases:

- ☒ AdventureWorks2022
- ☐ master
- ☐ model

☐ Ignore databases where the state is not online

OK Cancel

Change... Cancel

Maintenance Plan Wizard

### Select Report Options

Select options for saving or distributing a report of the maintenance plan actions.

☒ Write a report to a text file

Folder location:  ...

☐ E-mail report

To:

Help < Back Next > Finish >> Cancel

Maintenance Plan Wizard

### Complete the Wizard

Verify the choices made in the wizard, and then click Finish.

Click Finish to perform the following actions:

- Maintenance Plan Wizard
  - Create Maintenance Plan 'MaintenancePlan'
  - Define Back Up Database (Full) Task
  - Selected reporting options

Help < Back Next > Finish Cancel

Maintenance Plan Wizard

### Maintenance Plan Wizard Progress

Click Stop to interrupt the operation.

**Success** 5 Total 0 Error  
5 Success 0 Warning

Details:

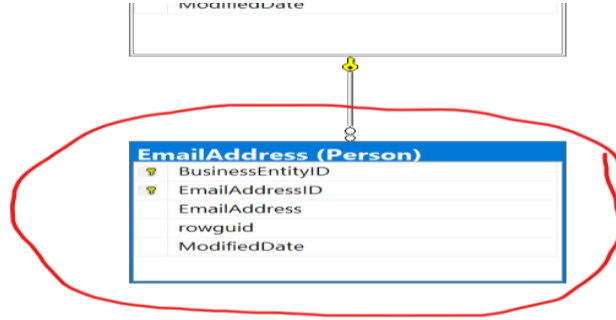
Action	Status	Message
Creating maintenance plan "MaintenancePlan"	Success	
Adding tasks to the maintenance plan	Success	
Adding scheduling options	Success	
Adding reporting options	Success	
Saving maintenance plan "MaintenancePlan"	Success	

Stop Report

Close

### c. Felaketten Kurtarma Senaryoları:

Burada yanlışlıkla Person.EmailAddress table'ını sildiğimiz senaryoyu ele alıyoruz. Normal şartlarda kendisinden pk alınan bir tabloyu silemeyiz ama bu senaryoda herhangi bir fk bağımlılığı olmayan Person.EmailAddress tablosunu siliyoruz.



Tabloyu siliyoruz.

```
Legacies\SQLXPRES...s2022 - Diagram_0*  SQLQu
DROP TABLE Person.EmailAddress;
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-23T00:25:47.2161830+03:00

```
SELECT * FROM Person.EmailAddress
```

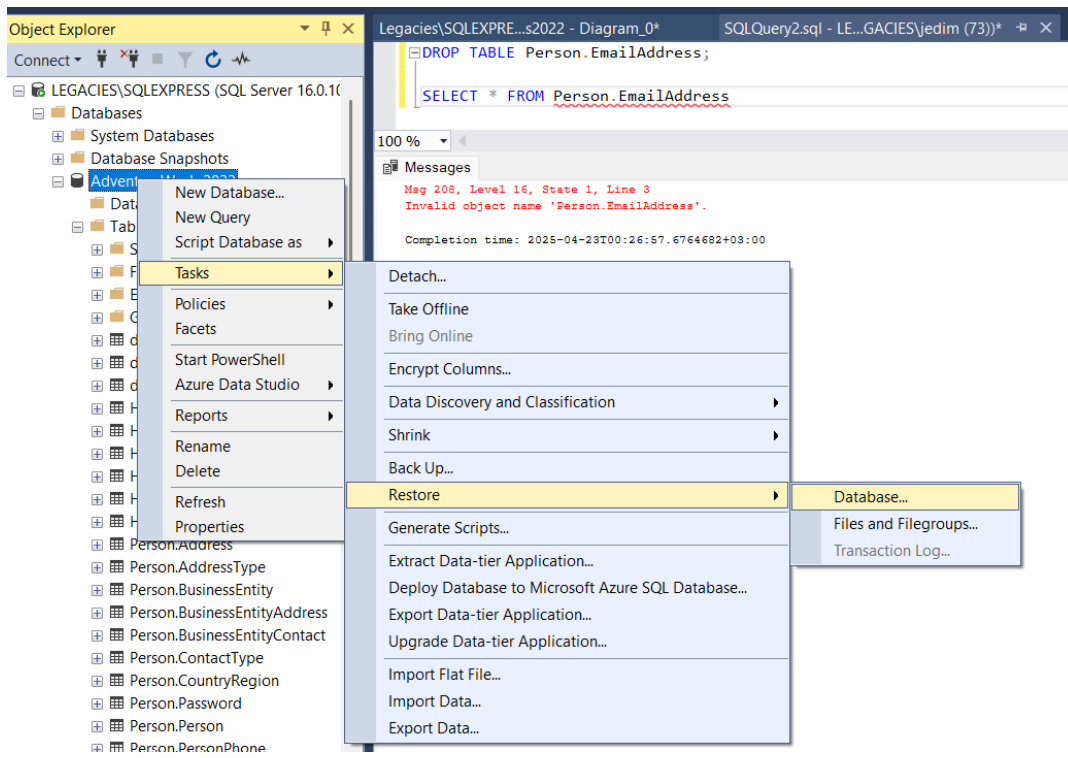
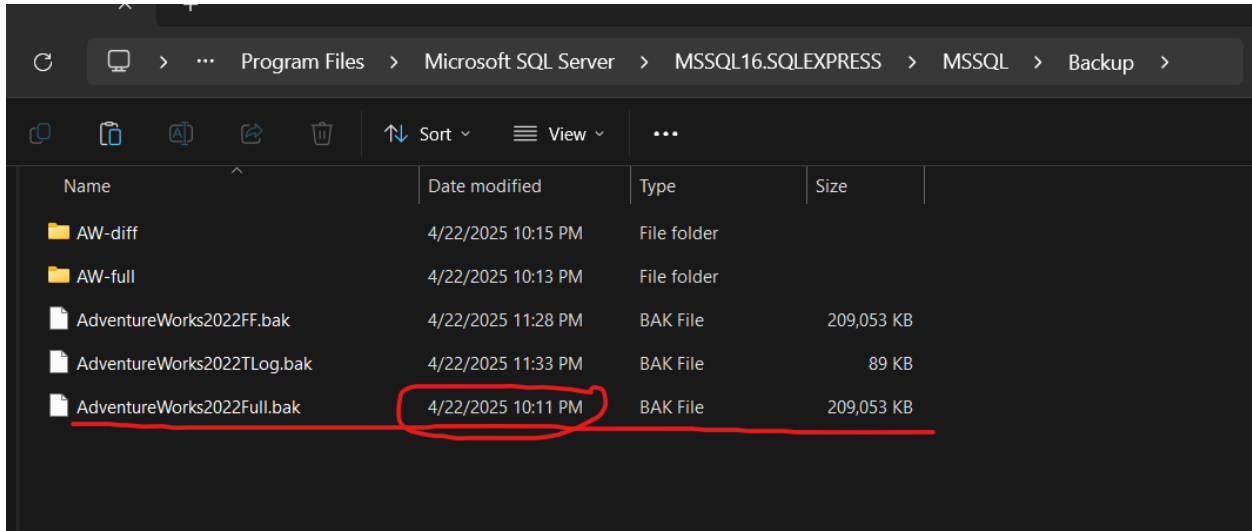
100 %

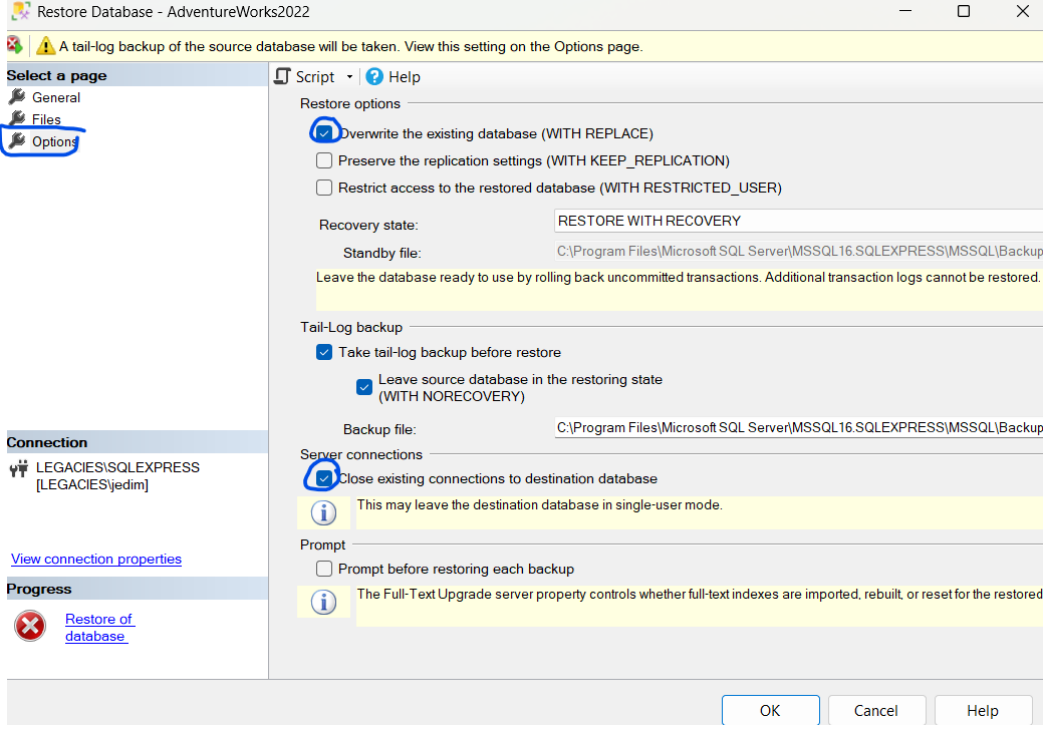
Messages

Msg 208, Level 16, State 1, Line 3  
Invalid object name 'Person.EmailAddress'.

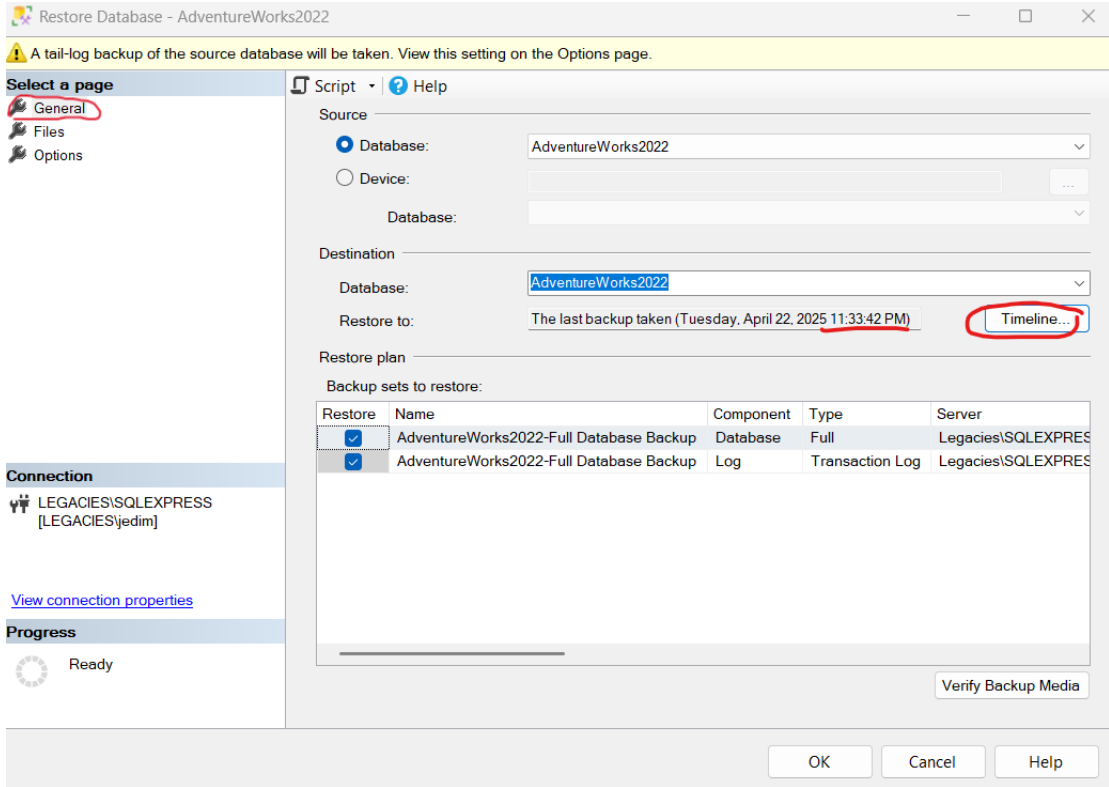
Completion time: 2025-04-23T00:26:57.6764682+03:00

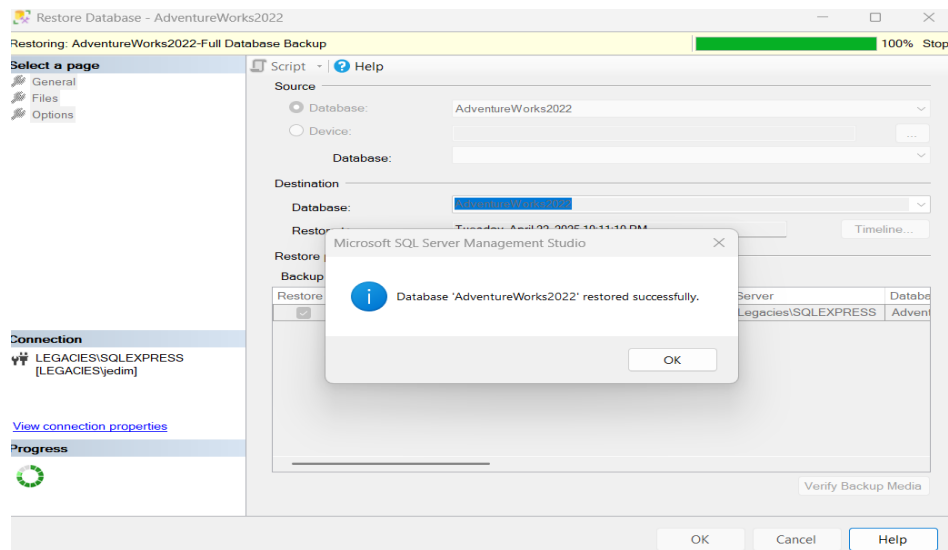
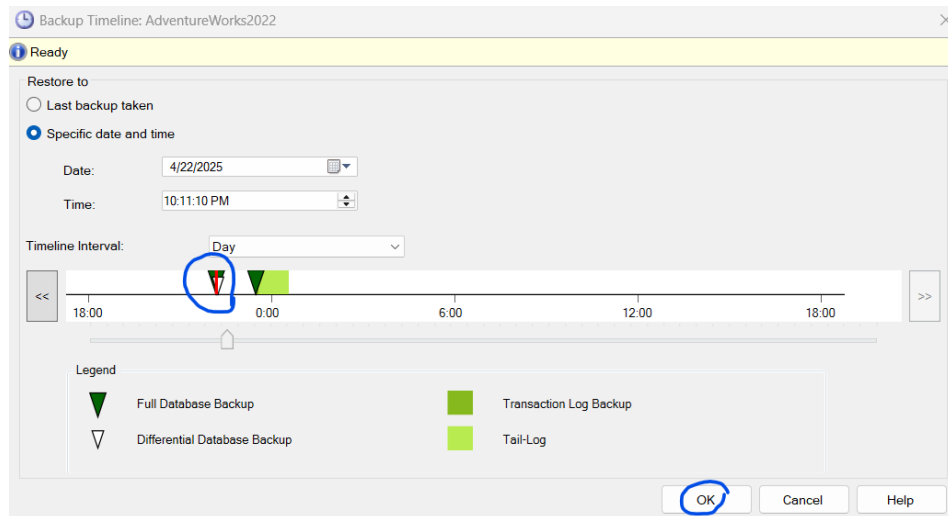
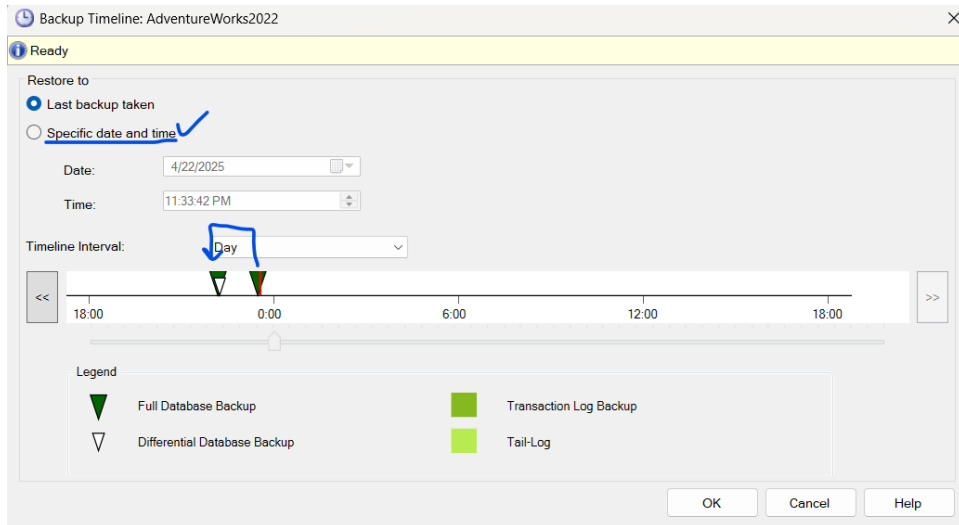
Silinen tabloyu geri getirmek için backup yaptığımız dosyadan restore ediyoruz.



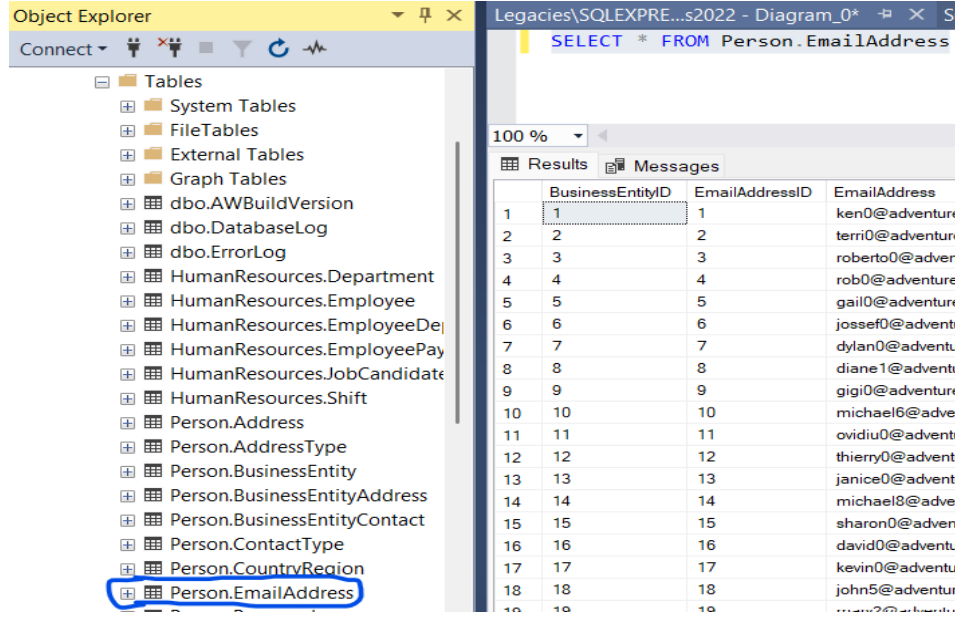


Yukarıdaki seçeneklerden, destination database'ın bağlantısını kopartmazsak hata alırız.





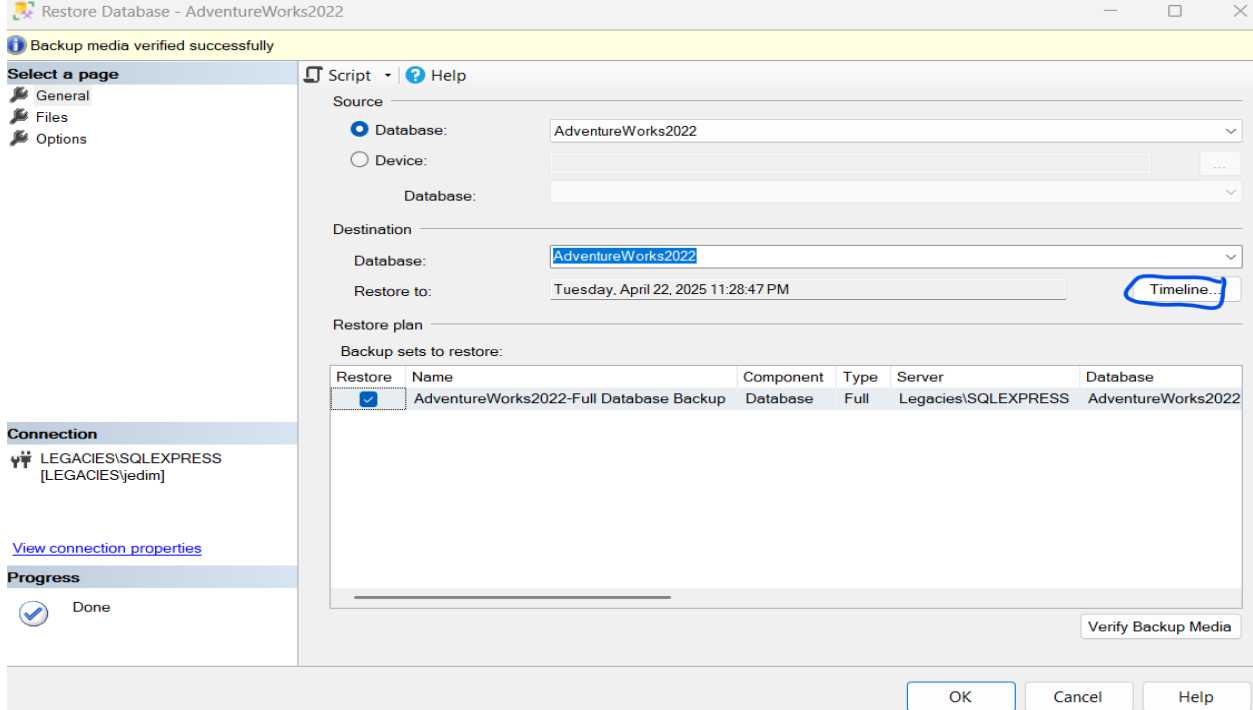




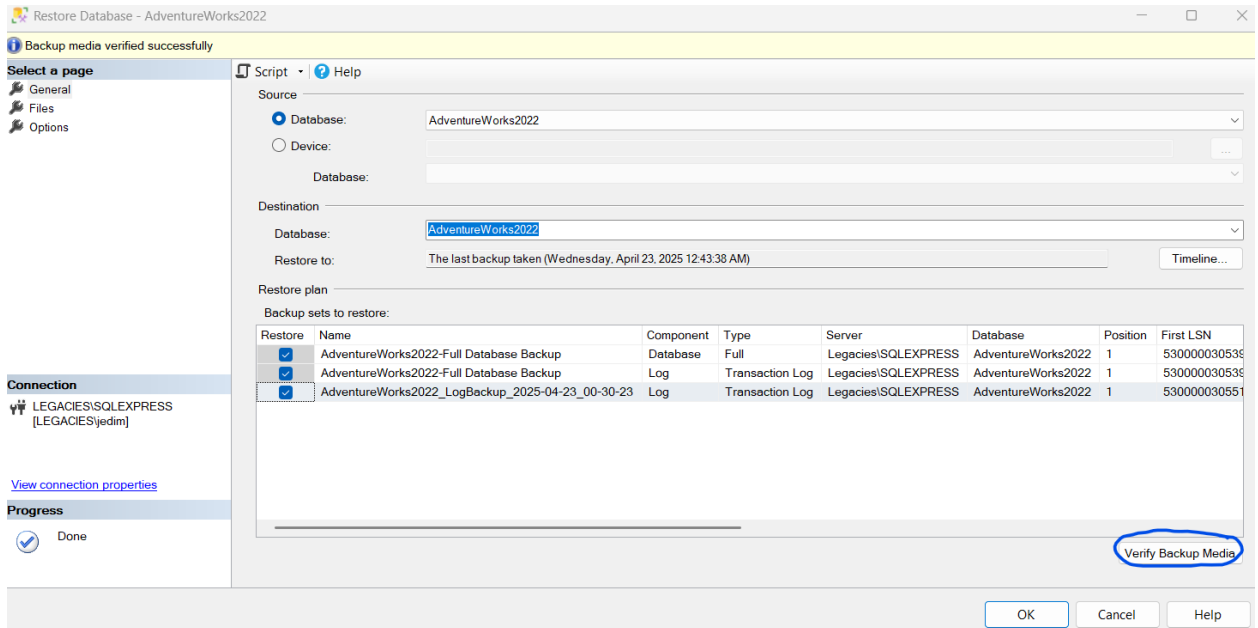
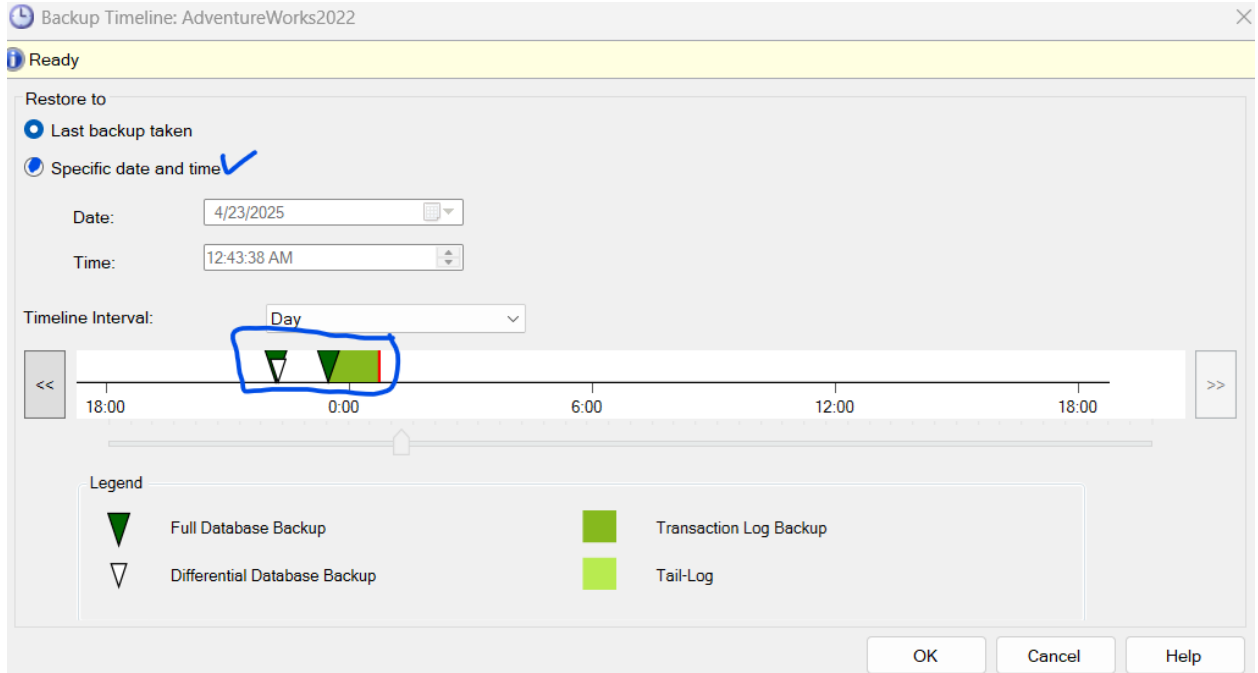
#### d. Test Yedekleme Senaryoları:

Bu kısımda hem sql kodu ile hem de ssms'in ui'ı ile nasıl yapılacağını ele alacağız.

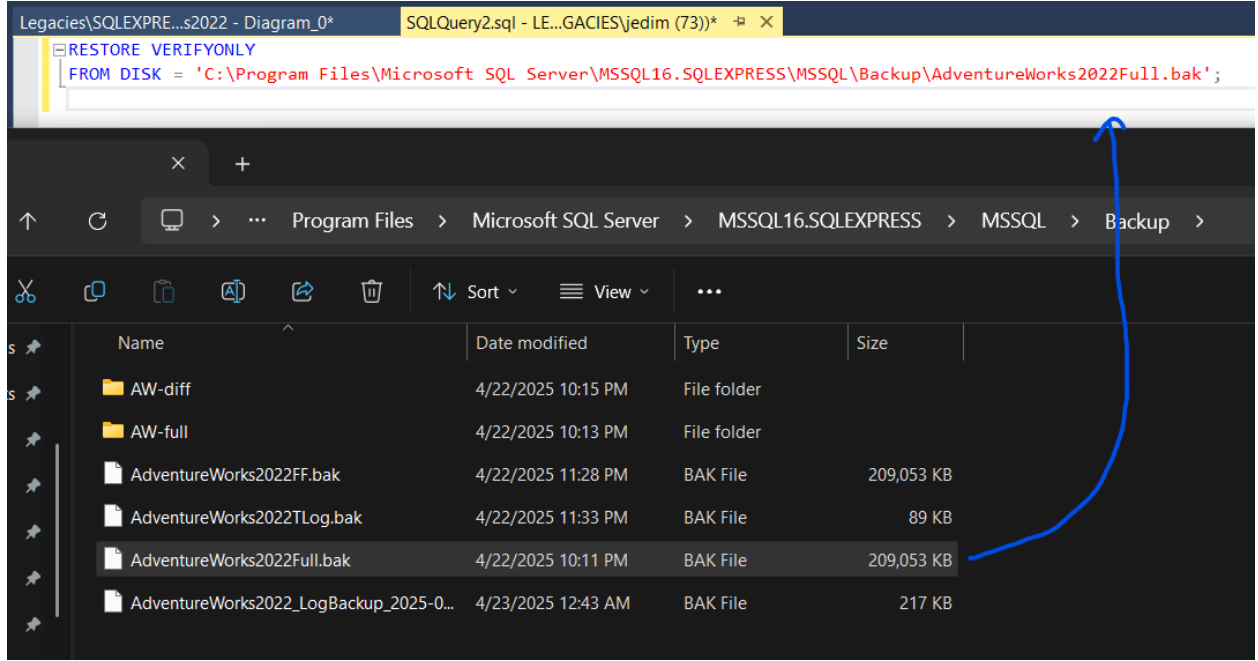
Database'e sağ tıklayıp backup kısmına tıklayacağız.



Verify etmek istediğimiz database backup'ını ve üzerinde daha önce yapılmış backup geçmişini seçip verify edebiliriz.



Buraya kadar olan kısım ssms'in ui'ı ile öncesinde backup ettiğimiz database'in verify edilmesini içeriyordu. Şimdi ise kod ile nasıl yapılacağını ele alacağız. Kod ile yapması çok daha hızlı ve basittir.



### 3. Veritabanı Güvenliği ve Erişim Kontrolü

#### a. Erişim Yönetimi:

Bu kısmı, [“Veritabanı Performans Optimizasyonu ve İzleme”](#) projesinin [“Veri Yöneticisi Roller”](#) kısmında ele aldığımız için burada tekrardan yazmaya gerek yoktur. Bu kısım ile ilgili her şey, ilk projede detaylıca ele alınmıştır.

#### b. Veri Şifreleme:

```
/* https://learn.microsoft.com/en-us/sql/relational-databases/security/  
USE master;  
GO  
  
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<UseStrongPasswordHere>';  
GO
```

Veritabanını şifrelemeden önce ilk aşamada, master veritabanında bir veritabanı anahtarı diğer bir deyişle Database Master Key (DMK) oluşturmamız. Bu master key, sertifikaları ve şifrelemek için kullanılacak anahtarı korumak için ön koruma sağlar.

```
CREATE CERTIFICATE MyServerCert
    WITH SUBJECT = 'My DEK Certificate';
GO
```

İkinci aşama olarak master veritabanı içinde bir sertifika oluşturulur. Bu sertifika, veriyi şifrelemek için kullanılacak anahtarı şifrelemeyi sağlayacak.

```
USE AdventureWorks2022;
GO

CREATE DATABASE ENCRYPTION KEY
    WITH ALGORITHM = AES_256
    ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
GO
```

Şifreleme yapmak istediğimiz database için, bir veritabanı şifreleme anahtarı oluşturur. AES'in 128, 192, 256 bitlik şifreleme seçeneklerinden en güçlü olan 256 bitlik olanı seçiyoruz ve öncesinde oluşturduğumuz sertifika ile bu anahtarı birbirine bağlıyoruz. Böylece veritabanı şifreleme anahtarı (DEK) sertifika ile korunmuş oluyor.

```
ALTER DATABASE AdventureWorks2022
    SET ENCRYPTION ON;
GO
```

Oluşturduğumuz DEK ile veritabanını şifreliyoruz.

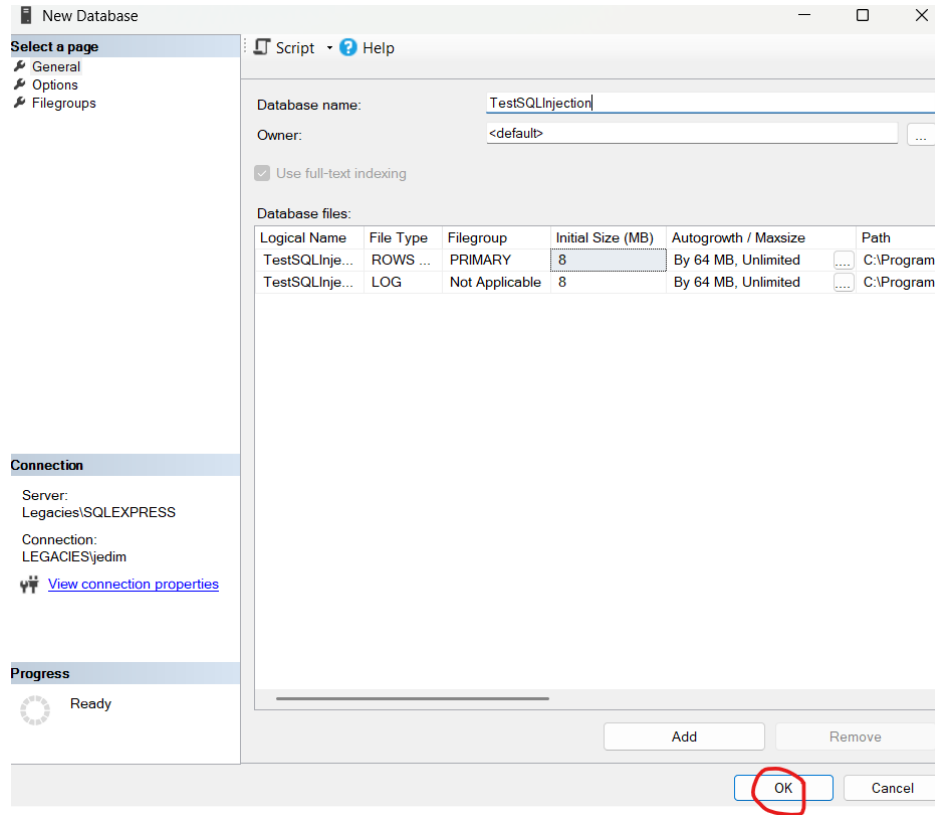
```
/* https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/verify-database-encryption */
USE AdventureWorks2022;
GO

/* The value 3 represents an encrypted state
on the database and transaction logs. */
SELECT *
FROM sys.dm_database_encryption_keys
WHERE encryption_state = 3;
```

Şifreleme durumunu görmek için microsoft'un sql server encryption web sitesindeki sql komutlarını kullanarak veritabanının doğru bir şekilde şifrenip şifrenmediğini gözlemliyoruz.

### c. SQL Injection Testleri:

Normal şartlarda select komutu dışındaki diğer komutlar ile uygulanabilecek sql injection saldırısına karşı herhangi bir koruması yoktur. Select komutu dışındaki komutları ise constraint ekleyerek engelleyebilmektedir. Select kısmından gelecek bir saldırının MSSQL veritabanı düzeyinde, dışarıdan log kayıtlarını izleyip müdahale edilmediği sürece otomatik olarak engellenebilmesi pek mümkün değildir. Bu yüzden, sql injection saldırısına karşı client bazlı bir koruma yapılması gerekmektedir. Normal sorgular yerine parametrelili sorgular kullanarak saldırganın, query'lerin açıklarını kullanmasının önüne geçilir. SQL injection kısmı için bir test ortamı oluşturulmuştur. Yeni bir veritabanı oluşturulup içerisine mock user data'ları (yapay kullanıcı verileri) eklenmiştir. Flask üzerinden bir client hazırlanmış olup parametresiz ve parametrelili sorgulardaki farklar ele alınmıştır.



TestSQLInjection isminde bir veritabanı oluşturuldu.

```
SQLQuery2.sql - LE...GACIES\jedim (78))* X SQLQuery1.sql - LE...GACIES\jedim (73))*  
CREATE TABLE Users (  
    UserID INT IDENTITY PRIMARY KEY,  
    Username NVARCHAR(50) NOT NULL,  
    Password NVARCHAR(50) NOT NULL,  
    Email NVARCHAR(100),  
    Role NVARCHAR(20),  
);  
  
INSERT INTO Users (Username, Password, Email, Role)  
VALUES  
    ('admin', 'admin123', 'admin@example.com', 'admin'),  
    ('user1', 'userpass1', 'user1@example.com', 'user'),  
    ('user2', 'userpass2', 'user2@example.com', 'user'),  
    ('user3', 'userpass3', 'user3@example.com', 'user'),  
    ('guest', 'guestpass', 'guest@example.com', 'user'),  
    ('sysadmin', 'sys123', 'sys@example.com', 'admin');
```

100 %  
Messages  
(6 rows affected)  
Completion time: 2025-04-23T14:34:04.4565588+03:00

Örnek bir user tablosu ve mock user verileri oluşturuldu. Password bilgileri, bilerek hashlenmemiş bir şekilde tutuldu. Best case'de kesinlikle password'ları veritabanlarında hashlenmiş bir şekilde tutulması gerekmektedir ama saldırganların genellikle password yerine veritabanındaki hassas bilgilere erişmek istediğinden ötürü bu şekilde test ortamı oluşturuldu.

```
SQLQuery4.sql - LE...GACIES\jedim (79))* X SQLQuery1.sql - LE...GACIES\jedim (73))*  
SELECT * FROM Users;
```

100 %  
Results Messages

	UserID	Username	Password	Email	Role
1	1	admin	admin123	admin@example.com	admin
2	2	user1	userpass1	user1@example.com	user
3	3	user2	userpass2	user2@example.com	user
4	4	user3	userpass3	user3@example.com	user
5	5	guest	guestpass	guest@example.com	user
6	6	sysadmin	sys123	sys@example.com	admin

Tablonun başarılı bir şekilde oluşturulup oluşturulmadığını test etmek için basit bir sorgu yazıldı.

## Parametresiz Query Senaryosu:

```
8 app = Flask(__name__)
9
10 # Veritabanı bağlantısı
11 connection_string = (
12     "Driver={ODBC Driver 17 for SQL Server};"
13     "Server=LEGACIES\\SQLEXPRESS;"
14     "Database=TestSQLInjection;"
15     "Trusted_Connection=yes;"
16 )
17
18 @app.route('/', methods=['GET', 'POST'])
19 def login():
20     result = ''
21     if request.method == 'POST':
22         username = request.form['username']
23         password = request.form['password']
24         query = f"SELECT * FROM Users WHERE Username = '{username}' AND Password = '{password}'"
25
26         try:
27             conn = pyodbc.connect(connection_string)
28             cursor = conn.cursor()
29             cursor.execute(query)
30             rows = cursor.fetchall()
31             if rows:
32                 result = f"Giriş başarılı! Kullanıcı: {rows}"
```

Problems Output Debug Console Terminal Ports Postman Console

```
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
tion WSGI server instead.
* Running on http://127.0.0.1:5000
```

Burada flask projesi yer almaktadır. Connection string değişkeni, ilgili veritabanına erişmek için yazıldı. Query değişkeninde ise parametresiz bir sorgu yazılmıştır. Client, 5000 portunda başlatıldı.

### Login

Username:

Password:

Giriş başarılı! Kullanıcı: [(1, 'admin', 'admin123', 'admin@example.com', 'admin'), (2, 'user1', 'userpass1', 'user1@example.com', 'user'), (3, 'user2', 'userpass2', 'user2@example.com', 'user'), (4, 'user3', 'userpass3', 'user3@example.com', 'user'), (5, 'guest', 'guestpass', 'guest@example.com', 'user'), (6, 'sysadmin', 'sys123', 'sys@example.com', 'admin')]

Görüldüğü üzere saldırgan username'e "OR 1=1" yazıp password kısmını boş bıraktığı zaman, client'teki parametresiz query'de şöyle gözükmektedir:

"SELECT \* FROM WHERE Username = '' OR 1=1 -- ' AND Password"

Sql'de comment line – işareti ile atıldığı için "AND Password" kısmı comment line içerisine alınmış olur. Böylece aslında tüm kullanıcı bilgileri getirilmiş olunur.

Bunu önlemenin birkaç basit yolu var. Username ve password data class'ında validation annotation kullanarak bu durum rahatlıkla çözülür. Bunlara literatürde middleware denir. Bir endpoint'e atılan post isteğinden gelen veriler, veritabanında işlem yapmayı sağlamadan önce middleware katmanı ile validate edilir. Diğer bir yöntem ise parametrelili yöntem kullanmaktır. Bu sayede, bu ve bunun gibi komutlarla sql syntax'ını manipüle edilemez durumda olur.

Parametrelili Query Senaryosu:

```
@app.route('/', methods=['GET', 'POST'])
def login():
    result = ''
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        #query = f"SELECT * FROM Users WHERE Username = '{username}' AND Password = '{password}'"
        query = "SELECT * FROM Users WHERE Username = ? AND Password = ?"
        try:
            conn = pyodbc.connect(connection_string)
            cursor = conn.cursor()
            cursor.execute(query, (username, password))
            rows = cursor.fetchall()
            if rows:
                result = f'Giriş başarılı! Kullanıcı: {rows}'
            else:
                result = 'Hatalı giriş!'
        except Exception as e:
            result = f'Hata oluştu: {e}'

    return render_template_string("""
        <h2>Login</h2>
```

Burada ise parametrelili bir query ile sql injection'a karşı bir koruma sağlandı.

## Login

Username:

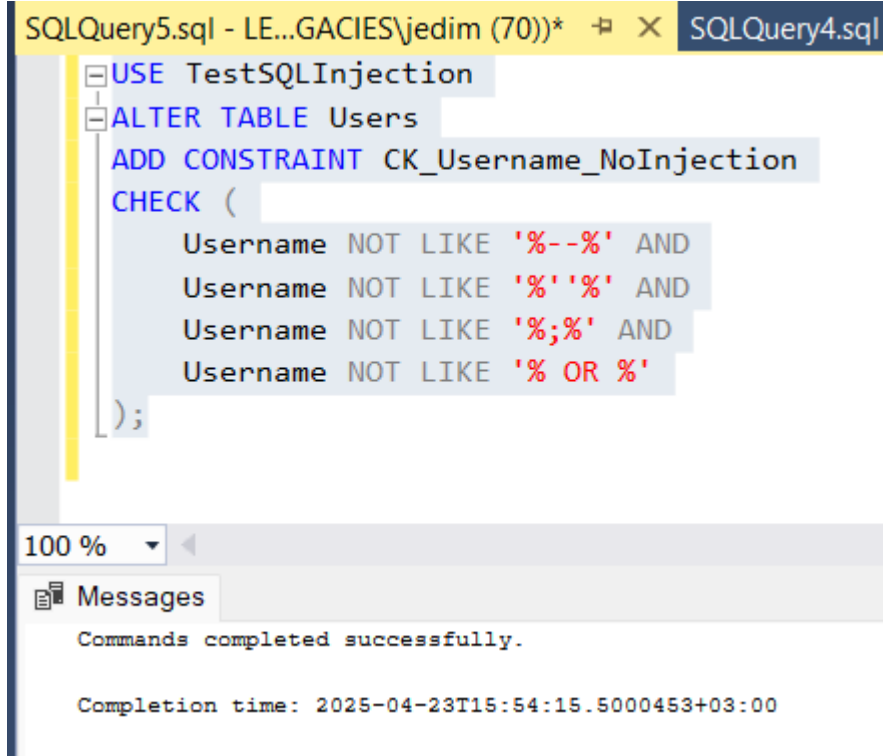
Password:

Login

Hatalı giriş!



Veritabanı düzeyinde sql injection komut türlerini Insert, Delete ve Update gibi işlemler için bloklamak için veritabanı için constraint açılabilir:



```
SQLQuery5.sql - LE...GACIES\jedim (70))* X SQLQuery4.sql
USE TestSQLInjection
ALTER TABLE Users
ADD CONSTRAINT CK_Username_NoInjection
CHECK (
    Username NOT LIKE '%--%' AND
    Username NOT LIKE '%''%' AND
    Username NOT LIKE '%;%' AND
    Username NOT LIKE '% OR %'
);
```

100 %

Messages

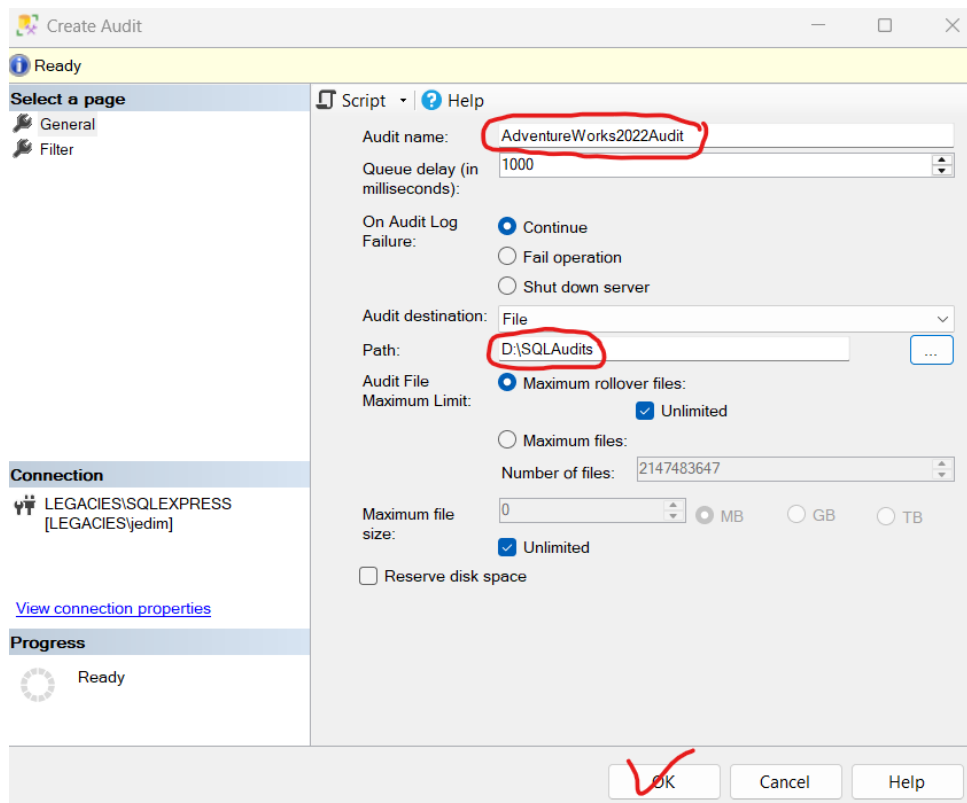
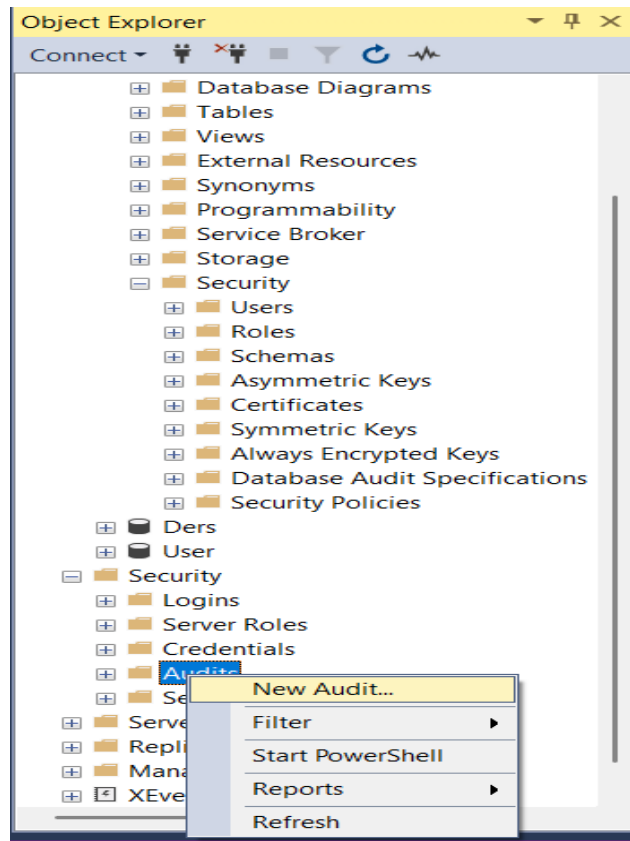
Commands completed successfully.

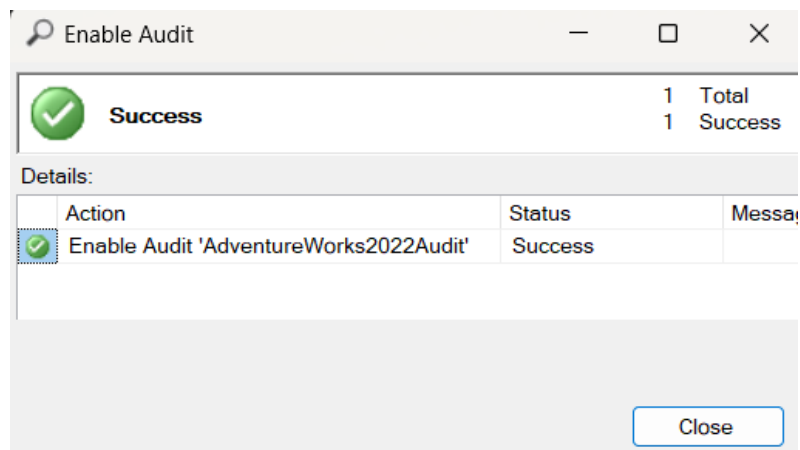
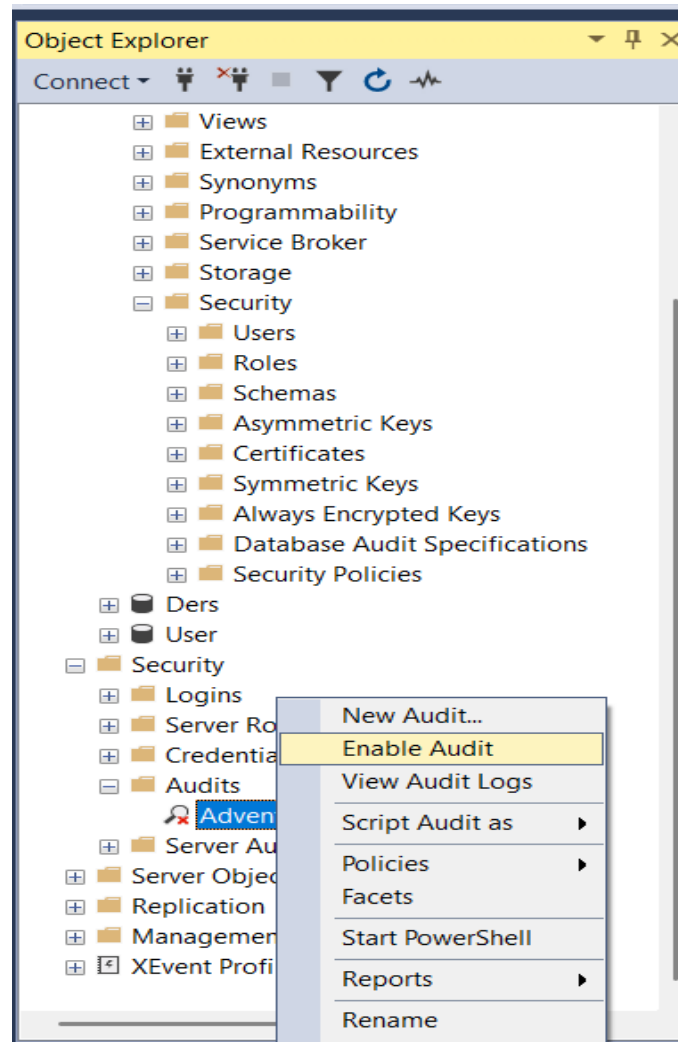
Completion time: 2025-04-23T15:54:15.5000453+03:00

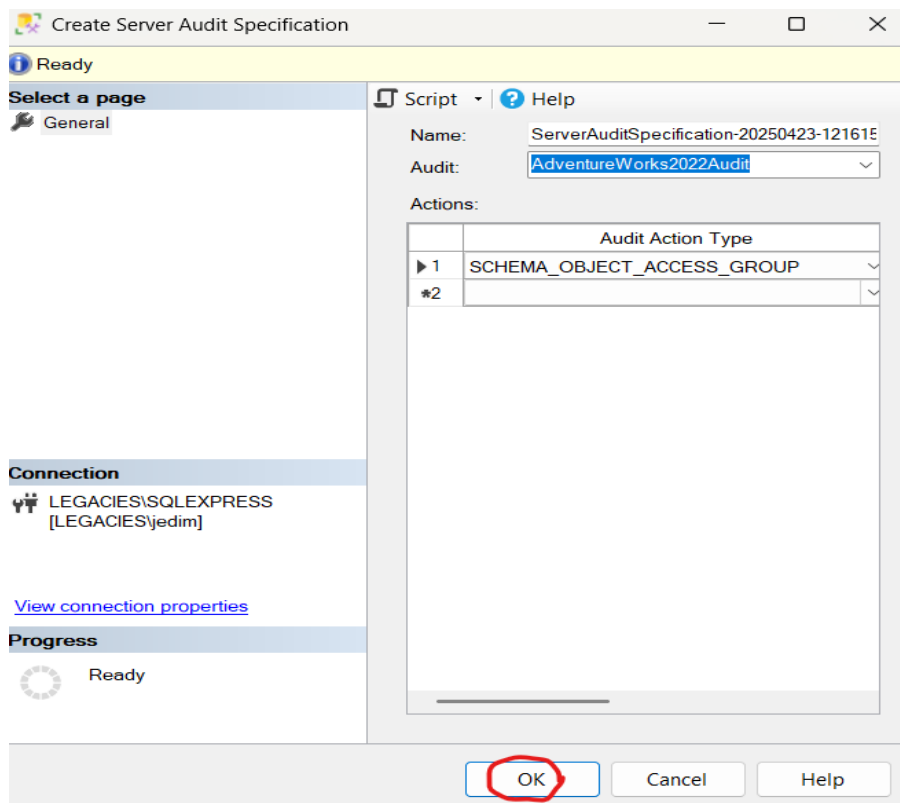
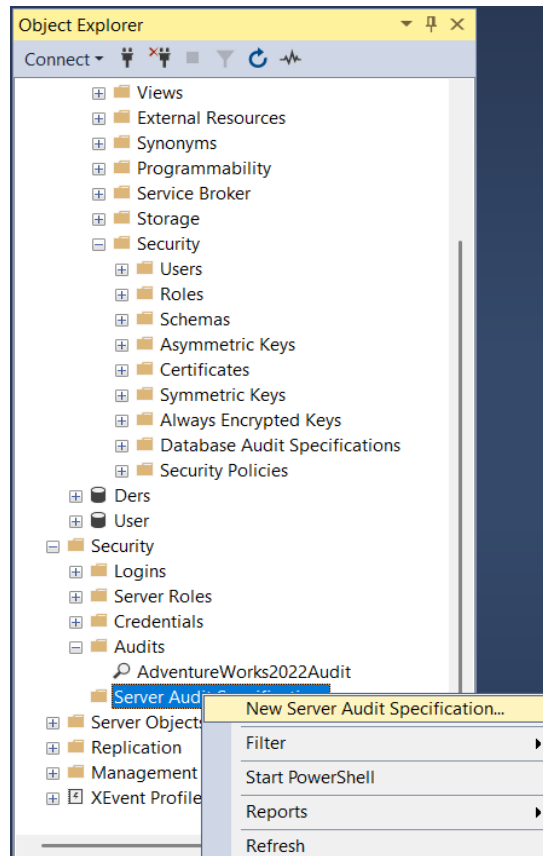
#### d. Audit Logları:

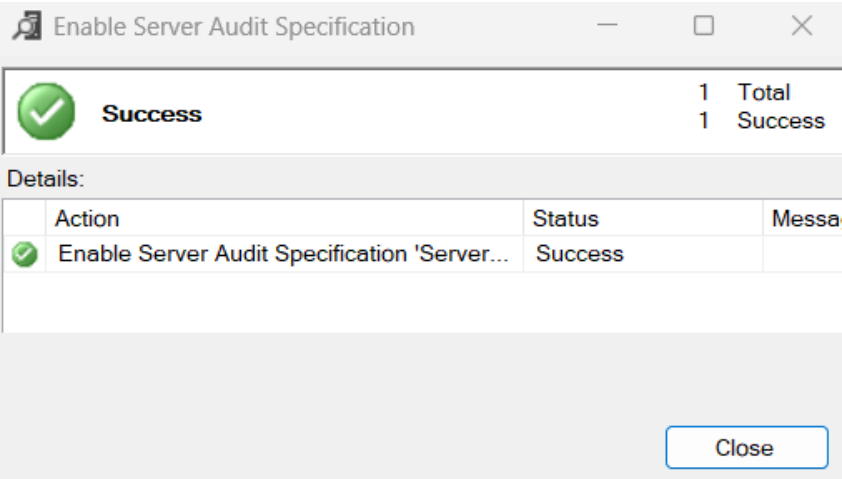
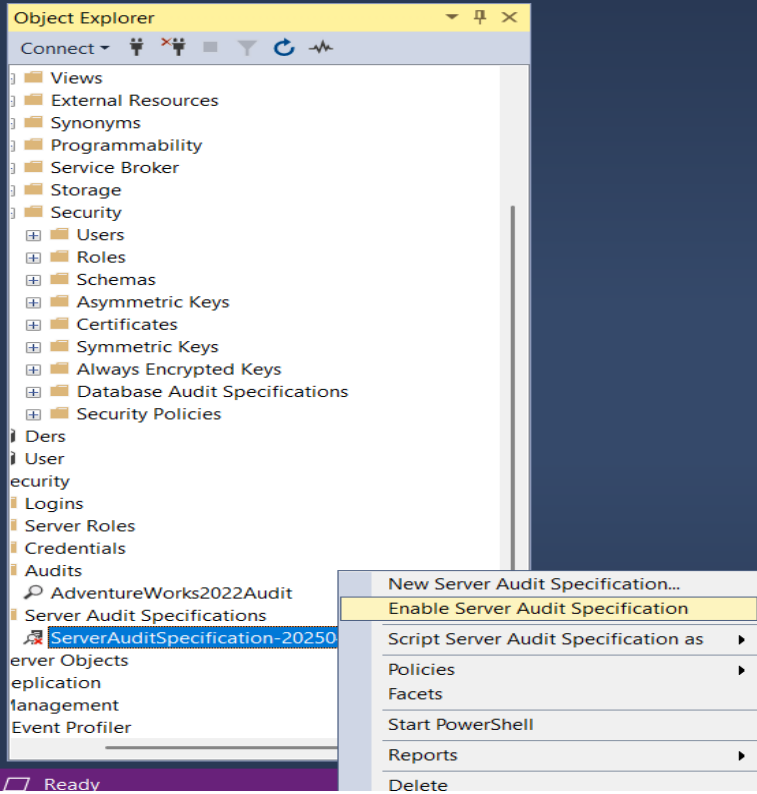
Audit logları ssms'teki kullanıcıların veritabanı/veritabanları üzerindeki aktivitesini gözlemlemek için kullanılır. Ssms'teki bazı audit log bilgileri,

Column Name	Description
Event time	Olayın gerçekleşme saati
Sequence number	Olay sırası
Permission bitmask	Yapılan işlemin izin tipi
Is column permission	Column seviyesinde mi işlem izni verildi?
Database principal id	Sorgu hangi yetki id'si ile veritabanında yapıldı?
Server principal id	Bağlantıyı başlatanın id'si









SQLQuery1.sql - LE...GACIES\jedim (73))\*

```
SELECT *
FROM sys.fn_get_audit_file('D:\SQLAudits\AdventureWorks2022Audit_9290E345-40DC-491E-BAAE-E5D8D259E754_0_133898732858930000.sqlaudit',
NULL,
NULL);
```

100 %

Results Messages

	event_time	sequence_number	action_id	succeeded	permission_bitmask	is_column_permission	session_id	server_principal_id	database_principal_id	target_sen
1	2025-04-23 09:14:45.9992023	1	AUSC	1	0x00000000000000000000000000000000	0	74	259	0	0
2	2025-04-23 09:19:02.1561275	1	SL	1	0x00000000000000000000000000000001	1	78	259	1	0
3	2025-04-23 09:19:02.1561275	1	SL	1	0x00000000000000000000000000000001	1	78	259	1	0
4	2025-04-23 09:19:02.1561275	1	SL	1	0x00000000000000000000000000000001	1	78	259	1	0
5	2025-04-23 09:19:02.1561275	1	SL	1	0x00000000000000000000000000000001	1	78	259	1	0
6	2025-04-23 09:19:02.1561275	1	SL	1	0x00000000000000000000000000000001	1	78	259	1	0
7	2025-04-23 09:19:02.1561275	1	SL	1	0x00000000000000000000000000000001	1	78	259	1	0
8	2025-04-23 09:21:50.3406771	1	SL	1	0x00000000000000000000000000000001	1	59	259	1	0
9	2025-04-23 09:21:50.3497835	1	SL	1	0x00000000000000000000000000000001	1	67	259	1	0
10	2025-04-23 09:22:46.3905517	1	SL	1	0x00000000000000000000000000000001	1	71	259	1	0
11	2025-04-23 09:22:46.4000953	1	SL	1	0x00000000000000000000000000000001	1	72	259	1	0
12	2025-04-23 09:22:48.5312250	1	SL	1	0x00000000000000000000000000000001	1	73	259	1	0
13	2025-04-23 09:22:48.5475683	1	SL	1	0x00000000000000000000000000000001	1	73	259	1	0
14	2025-04-23 09:22:48.5530863	1	EX	1	0x00000000000000000000000000000020	0	73	259	1	0