

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



AĞ TABANLI PARALEL DAĞITIM SİSTEMLERİ PROJE RAPORU

Berfin Savran
21290637

26.04.2025

<https://github.com/BerfinSavran>

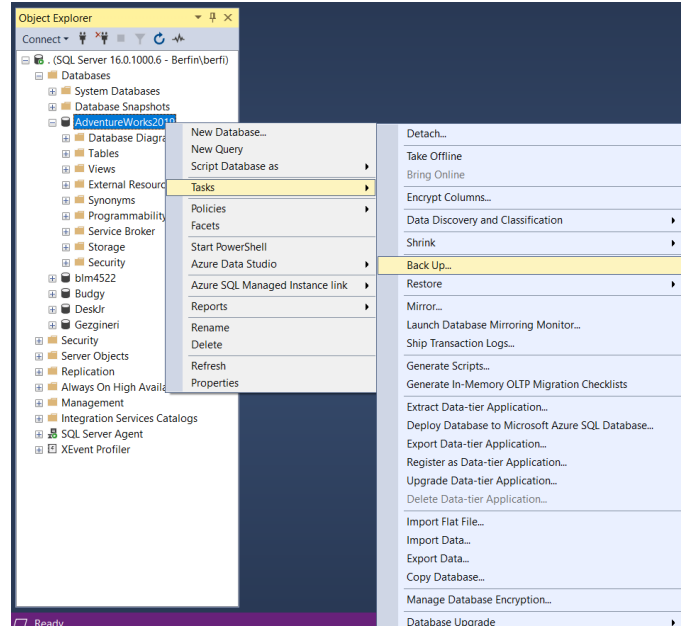
PROJE 7: VERİTABANI YEDEKLEME VE OTOMASYON ÇALIŞMASI

Adım 1: Sql Server Agent çalışıyor olmalı.

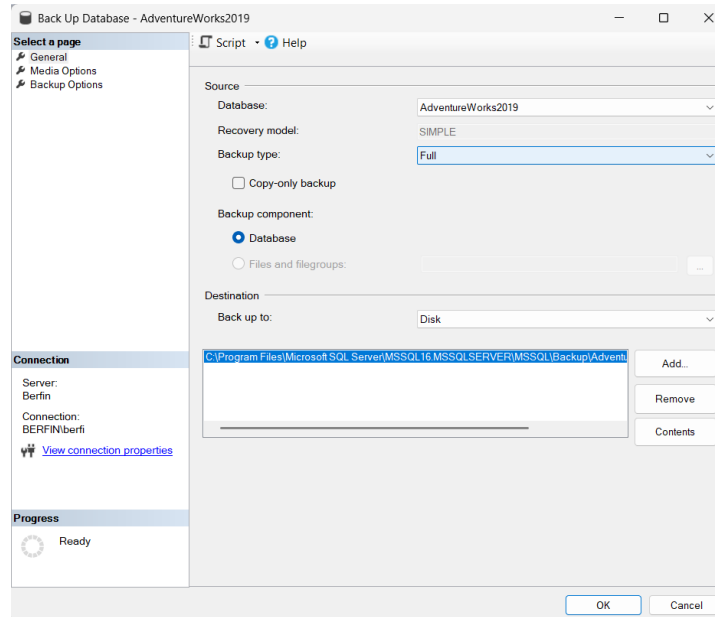
- SSMS → Object Explorer → SQL Server Agent → Yeşil mi? Eğer kapalıysa sağ tıklayıp "Start" yapıyoruz.

Adım 2: Manuel Yedekleme Alma

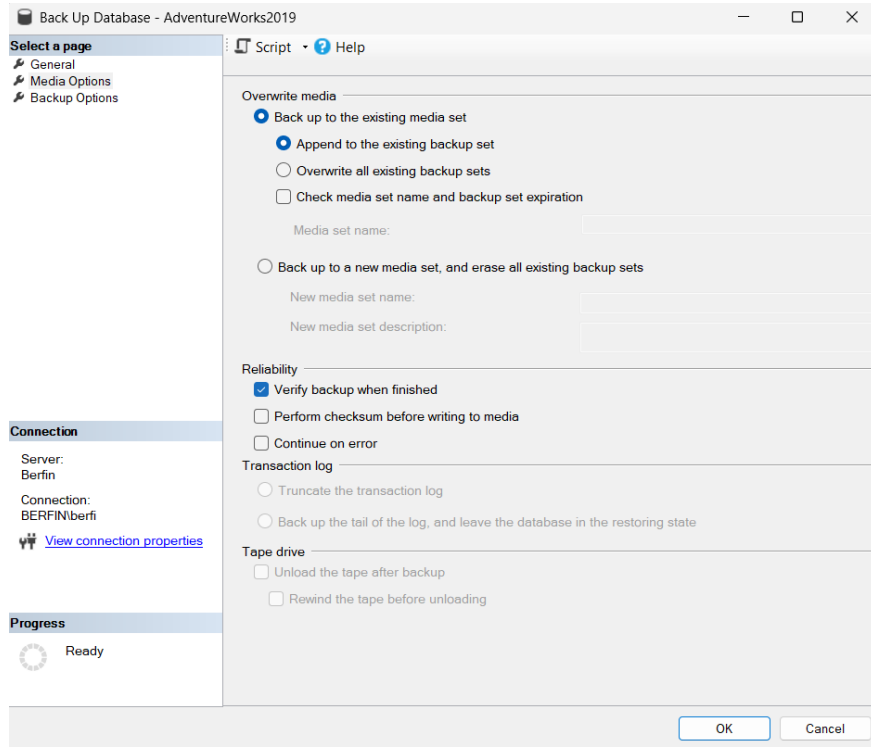
- SSMS'de AdventureWorks veritabanına sağ tıklayalım.
- Tasks > Back Up... seçeneğini seçelim.



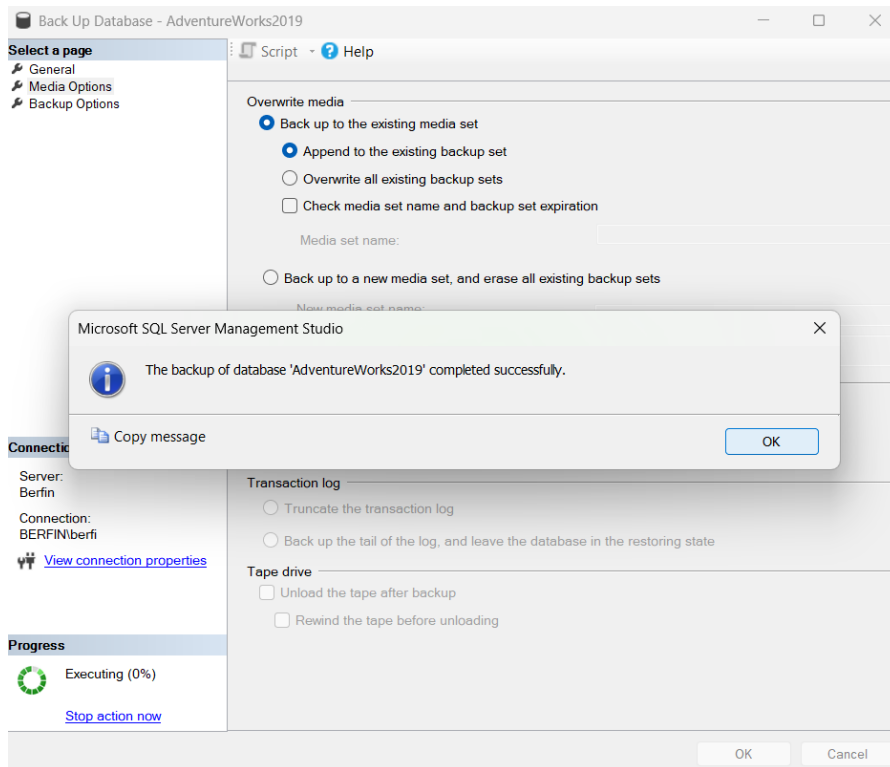
- Backup type: Full olarak ayarlayalım.
- Destination bölümünde varsayılan konumu kullanabiliriz veya yeni bir konum ekleyebiliriz.



- Options sayfasında "Verify backup when finished" seçeneğini işaretleyelim.

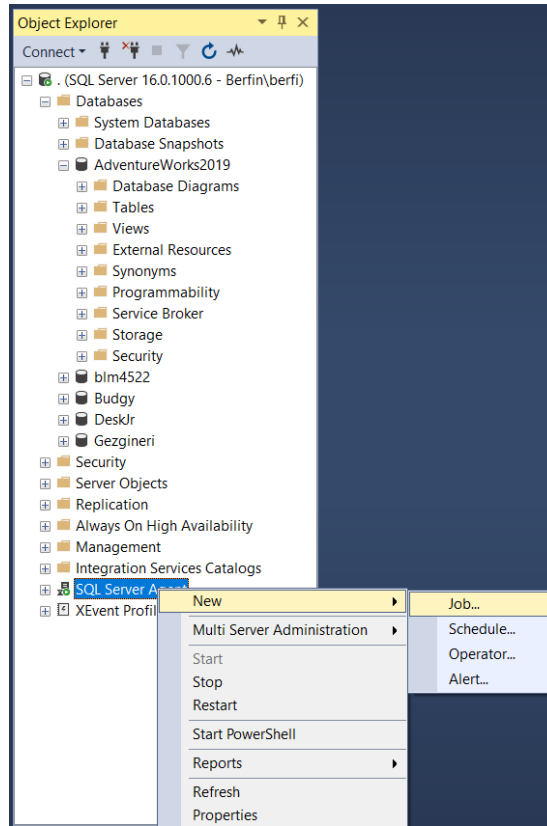


- OK'e tıklayarak tam yedeği alabiliriz.

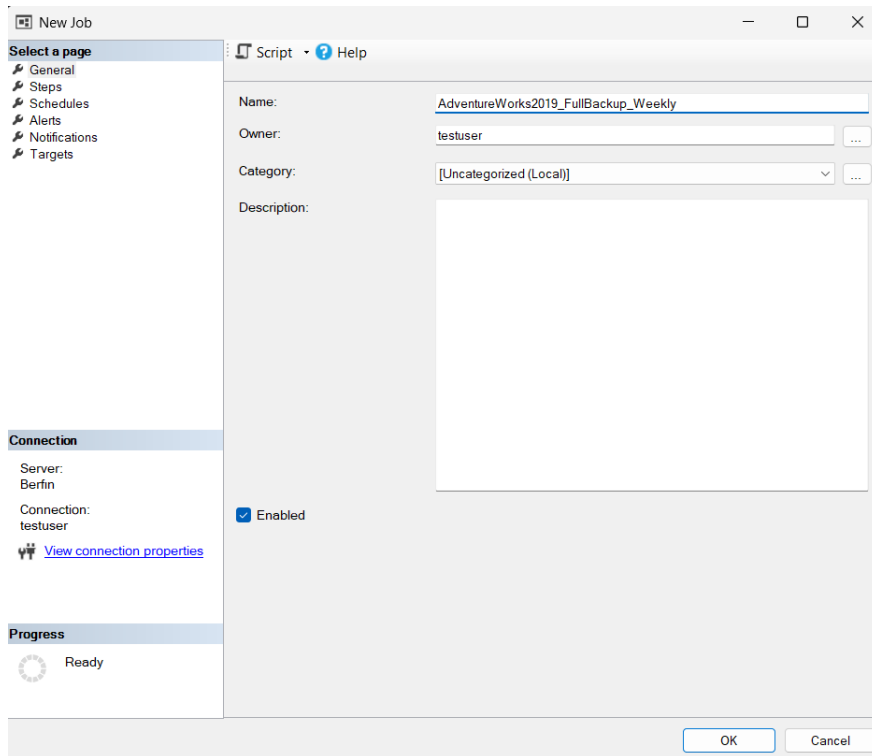


Adım 3: Full Backup Almak (Tam Yedekleme)

- SQL Server Agent'a sağ tıklayalım ve "New > Job" seçeneğini seçelim.



- Genel sekmesinde iş için bir ad verelim
"AdventureWorks2019_FullBackup_Weekly"



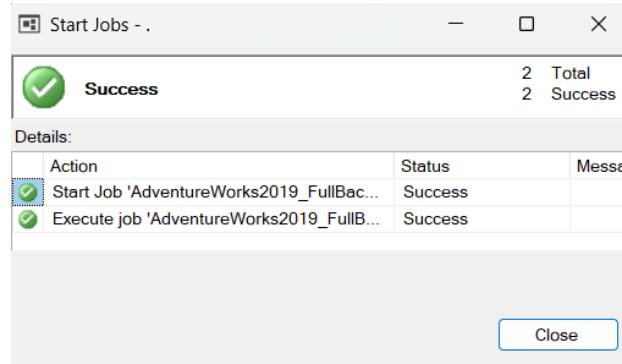
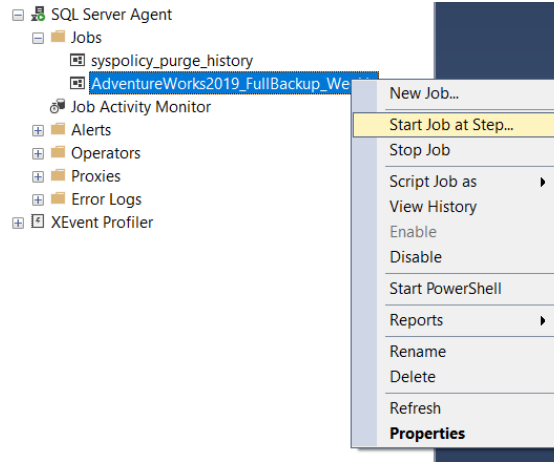
- Steps sekmesinde "New" düğmesine tıklayalım.
- Step adı olarak "Full Backup" girelim.

- Type olarak "Transact-SQL script (T-SQL)" seçelim.
- T-SQL kodunu girelim ve OK tıklayalım.

- Schedules sekmesine gidedim ve "New" düğmesine tıklayalım.
- Haftalık bir zamanlama oluşturalım (her Pazar günü saat 00:00 olsun).

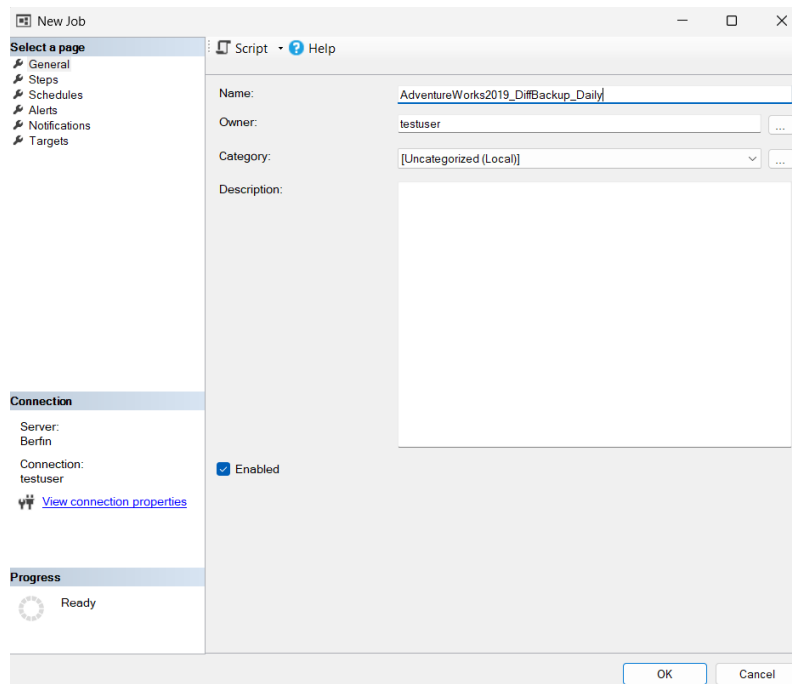
- OK'e tıklayarak programı kaydedelim.
- OK'e tıklayarak işi kaydedelim.

- Şimdi de test edelim.
- Oluşturduğumuz job üzerinde sağ tıklayıp "Start Job at Step" tıklayalım.



Adım 4: Fark Yedeklemesi İçin Job Oluşturma

- Aynı adımları tekrarlayalım.
- Job adı: "AdventureWorks2019_DiffBackup_Daily"



- Step adı: "Differential Backup"

New Job Step

Select a page

- General
- Advanced

Script ? Help

Step name: Differential Backup

Type: Transact-SQL script (T-SQL)

Run as:

Database: master

Command:

```
BACKUP DATABASE AdventureWorks2019  
TO DISK = C:\Program Files\Microsoft SQL Server\MSSQL16\MSSQLSERVER\MS  
WITH DIFFERENTIAL, INIT, NAME = 'AdventureWorks2019 Differential Backup', S
```

Open... Select All Copy Paste Parse

Connection

Server: Berfin

Connection: testuser

[View connection properties](#)

Progress

Ready

Previous Next

OK Cancel

- Zamanlama: Her gün (Pazartesi'den Cumartesi'ye, saat 20:00)

New Job Schedule

Name: Differential

Jobs in Schedule

Schedule type: Recurring

Enabled

One-time occurrence

Date: 26.04.2025 Time: 10:59:57

Frequency

Occurs: Weekly

Recurs every: 1 week(s) on

Monday Wednesday Friday Saturday

Tuesday Thursday Sunday

Daily frequency

Occurs once at: 20:00:00

Occurs every: 1 hour(s)

Starting at: 00:00:00

Ending at: 23:59:59

Duration

Start date: 26.04.2025

End date: 26.04.2025

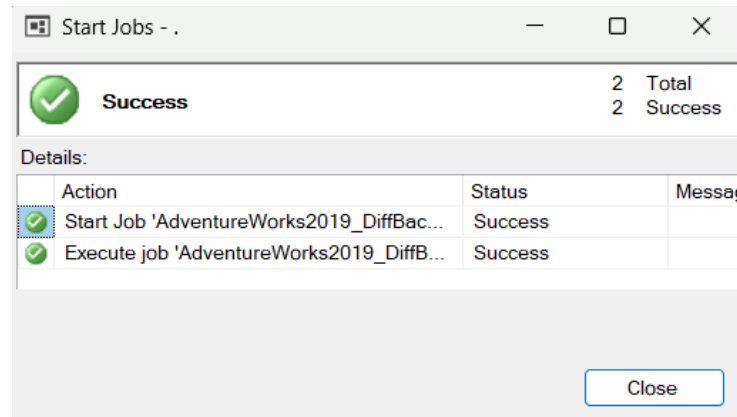
No end date

Summary

Description: Occurs every week on Monday, Tuesday, Wednesday, Thursday, Friday, Saturday at 20:00:00. Schedule will be used starting on 26.04.2025.

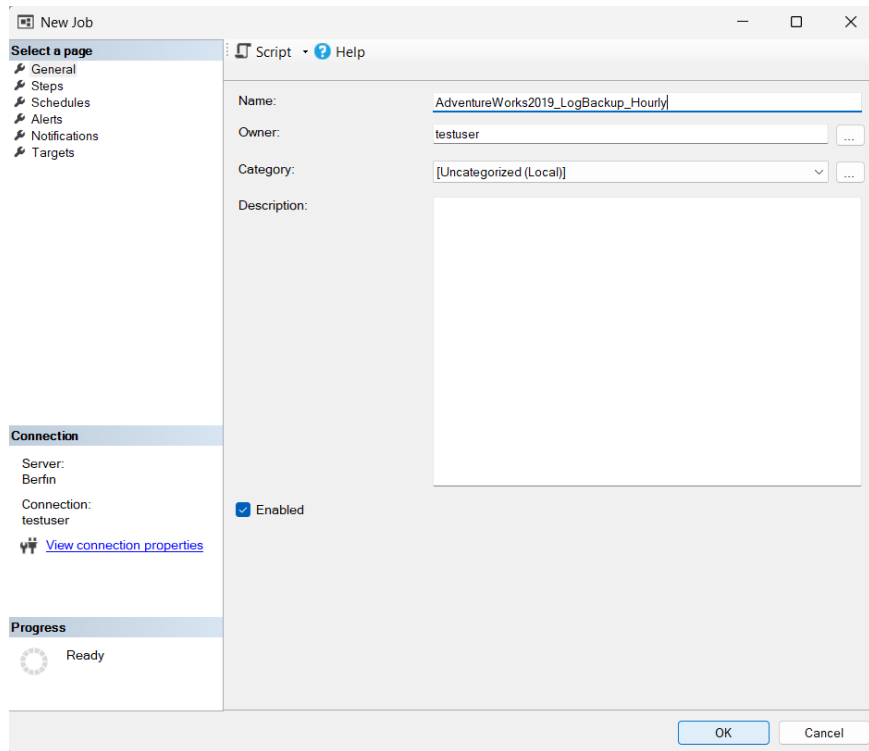
OK Cancel Help

- Test edelim.



Adım 5: İşlem Günlüğü Yedeklemesi İçin Job Oluşturma

- Yine aynı adımları tekrarlayalım.
- Job adı: "AdventureWorks2019_LogBackup_Hourly"



- Step adı: "Transaction Log Backup"

New Job Step

Select a page

- General
- Advanced

Script ? Help

Step name: Transaction Log Backup

Type: Transact-SQL script (T-SQL)

Run as:

Database: master

Command:

```
BACKUP LOG AdventureWorks2019  
TO DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\M...' WITH INIT, NAME = 'AdventureWorks2019 Hourly Transaction Log Backup', SKIP,
```

Open... Select All Copy Paste Parse

Server: Berlin
Connection: testuser
[View connection properties](#)

Progress

Ready

Previous Next

OK Cancel

- Zamanlama: Her saat (00-23, her gün)

New Job Schedule

Name: Daily

Schedule type: Recurring

Enabled

One-time occurrence

Date: 26.04.2025 Time: 11:10:03

Frequency

Occurs: Daily

Recurs every: 1 day(s)

Daily frequency

Occurs once at: 00:00:00

Occurs every: 1 hour(s)

Starting at: 00:00:00

Ending at: 23:00:00

Duration

Start date: 26.04.2025

End date: 26.04.2025

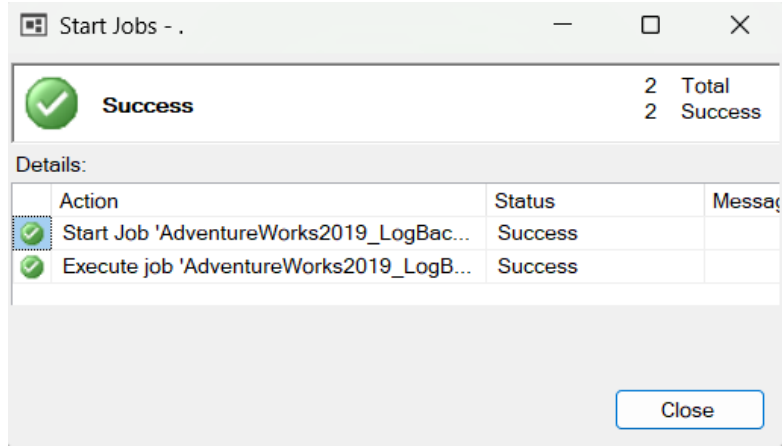
No end date:

Summary

Description: Occurs every day every 1 hour(s) between 00:00:00 and 23:00:00. Schedule will be used starting on 26.04.2025.

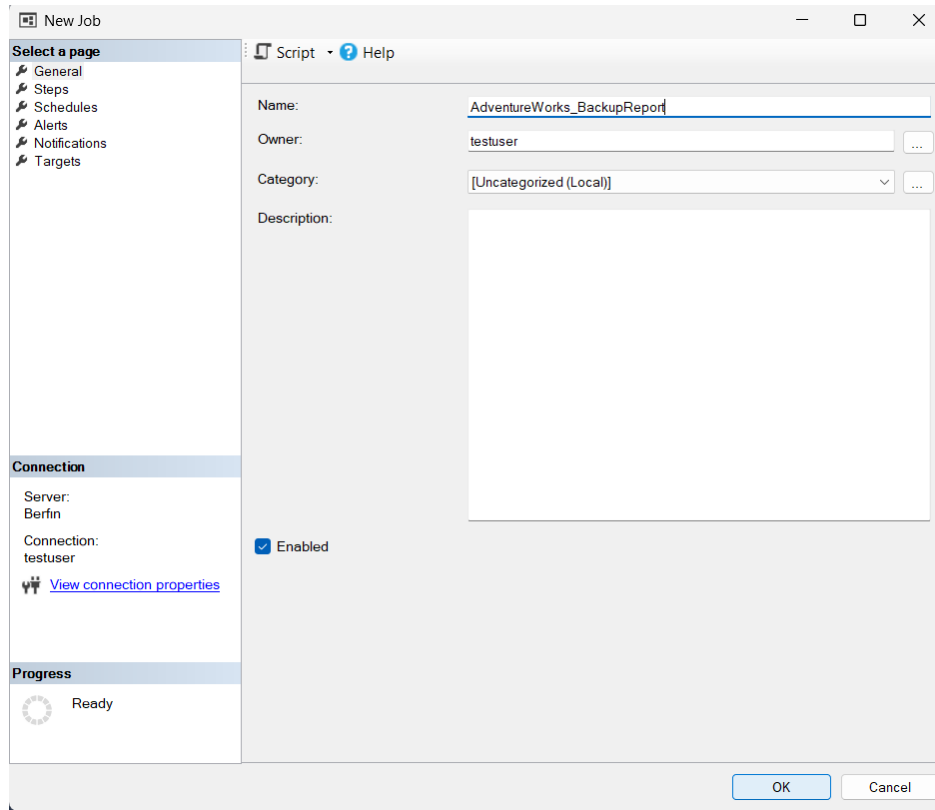
OK Cancel Help

- Öncesinde Full Backup alalım.
- Test edelim.



Adım 6: Yedekleme Raporlama Job'ı Oluşturma

- Job adı: "AdventureWorks_BackupReport"



- Step adı: "Generate Backup Report"

New Job Step

Select a page: General, Advanced

Script Help

Step name: Generate Backup Report

Type: Transact-SQL script (T-SQL)

Run as:

Database: master

Command:

```
SELECT
  database_name,
  backup_start_date,
  backup_finish_date,
  CASE backup_type
    WHEN 'D' THEN 'Full Database Backup'
    WHEN 'I' THEN 'Differential Database Backup'
    WHEN 'L' THEN 'Transaction Log Backup'
    ELSE 'Other'
  END AS backup_type,
  backup_size / 1024 / 1024 AS backup_size_MB,
  physical_device_name
FROM msdb.dbo.backupset bs
INNER JOIN msdb.dbo.backupmediafamily bmf ON bs.media_set_id = bmf.media_set_id
WHERE database_name = 'AdventureWorks2019'
ORDER BY backup_finish_date DESC;
```

Open... Select All Copy Paste Parse

Previous Next

OK Cancel

- Zamanlama: Her gün saat 09:00'da çalışsın.

New Job Schedule

Name: Backup Report Jobs in Schedule

Schedule type: Recurring Enabled

One-time occurrence

Date: 26.04.2025 Time: 11:26:45

Frequency

Occurs: Daily

Recurs every: 1 day(s)

Daily frequency

Occurs once at: 09:00:00

Occurs every: 1 hour(s) Starting at: 00:00:00 Ending at: 23:59:59

Duration

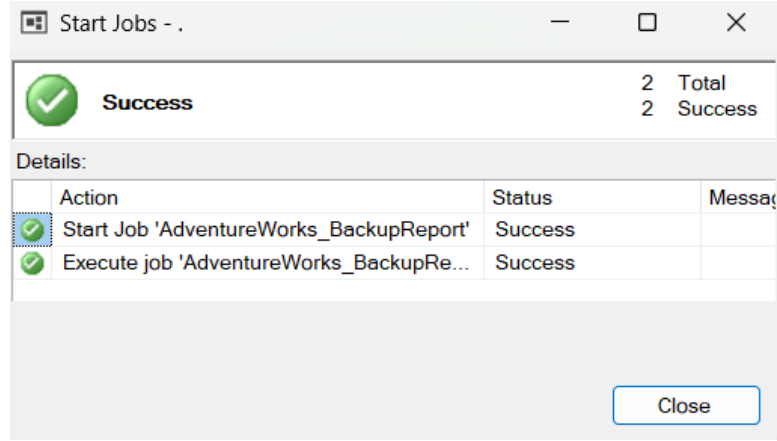
Start date: 26.04.2025 End date: 26.04.2025 No end date

Summary

Description: Occurs every day at 09:00:00. Schedule will be used starting on 26.04.2025.

OK Cancel Help

- Test edelim.

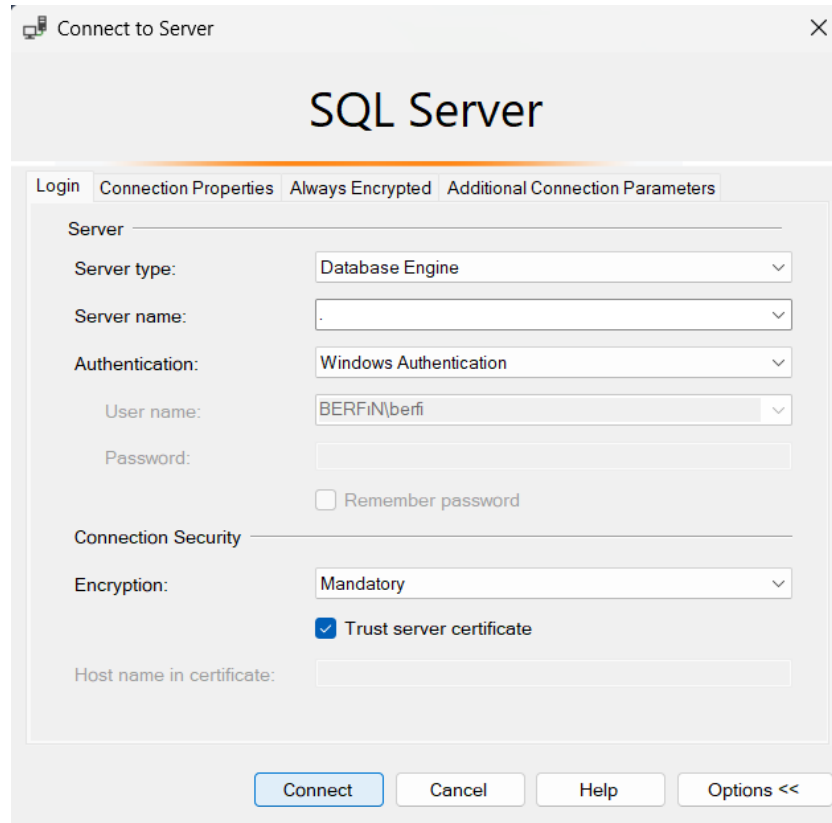


	database_name	backup_start_date	backup_finish_date	backup_type	backup_size_MB	physical_device_name
1	AdventureWorks2019	2025-04-26 11:21:11.000	2025-04-26 11:21:11.000	Transaction Log Backup	0.08203125000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...
2	AdventureWorks2019	2025-04-26 11:19:50.000	2025-04-26 11:19:51.000	Full Database Backup	201.08593750000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...
3	AdventureWorks2019	2025-04-26 11:03:54.000	2025-04-26 11:03:54.000	Differential Database Backup	2.08984375000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...
4	AdventureWorks2019	2025-04-26 10:54:23.000	2025-04-26 10:54:23.000	Full Database Backup	201.08984375000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...
5	AdventureWorks2019	2025-04-24 14:14:10.000	2025-04-24 14:14:10.000	Full Database Backup	199.92578125000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...
6	AdventureWorks2019	2025-04-22 11:37:43.000	2025-04-22 11:37:43.000	Full Database Backup	200.08593750000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...
7	AdventureWorks2019	2023-05-08 12:08:45.000	2023-05-08 12:08:45.000	Full Database Backup	199.08593750000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...

PROJE 3: VERİTABANI GÜVENLİĞİ VE ERIŞİM KONTROLÜ

Adım 1: Erişim Yönetimi: Kullanıcı Erişimi ve Yetkilendirme

- Windows Authentication ile giriş yapalım.



- Windows Authentication kullanıldığında:
 - SQL Server, kullanıcı kimliğini Windows işletim sistemi üzerinden doğrular.
(Yani Windows'a nasıl giriş yaptıysak, SQL Server o oturumu kullanır.)
 - Eğer Windows hesabımız SQL Server'da yetkiliyse, tekrar şifre yazmadan giriş yaparız.
 - Şifreler SQL Server'e taşınmaz. Kimlik doğrulama doğrudan Windows üzerinde yapılır. Daha güvenlidir.
(Saldırgan SQL Server'e erişse bile Windows şifresini bilemez.)
- SQL Server Authentication kullanarak yeni kullanıcı (Login) oluşturalım.
- Kullanıcıyı AdventureWorks2019 veritabanına ekleyelim.

```
CREATE LOGIN testuser WITH PASSWORD = 'StrongPassword123';  
CREATE USER testuser FOR LOGIN testuser;
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-22T13:11:35.2074824+03:00

- Giriş yapabiliyor muyuz test edelim.

Connect to Server

SQL Server

Login | Connection Properties | Always Encrypted | Additional Connection Parameters

Server

Server type: Database Engine

Server name: .

Authentication: SQL Server Authentication

Login: testuser

Password: *****

☐ Remember password

Connection Security

Encryption: Mandatory

☒ Trust server certificate

Host name in certificate:

Connect Cancel Help Options <<

Adım 2: Kullanıcıya Yetki Verme

- db_datareader: Veritabanındaki tüm verileri okuma yetkisi verelim.
- db_datawriter: Veritabanındaki tüm tablolara yazma (insert, update, delete) yetkisi verelim.

```
USE AdventureWorks2019;  
EXEC sp_addrolemember 'db_datareader', 'testuser'; -- Veritabanı okuma yetkisi verir
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-22T13:18:20.0053104+03:00

```
USE AdventureWorks2019;  
EXEC sp_addrolemember 'db_datawriter', 'testuser'; -- Veritabanı yazma yetkisi verir
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-26T11:55:32.7470611+03:00

Adım 3: Veri Şifreleme: Transparent Data Encryption (TDE)

- Master Key oluşturalım.
- Bu, ana anahtardır. Şifrelenmiş sertifikalar gibi üst düzey şifrelemeler için kullanılır.

```
USE master;  
GO  
  
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'StrongPassword123!';
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-22T13:20:46.7309387+03:00

- Sertifika oluşturalım.
- Bu sertifika, şifreleme işlemleri için kullanılacaktır.

```
CREATE CERTIFICATE TDECertificate
WITH SUBJECT = 'TDE Certificate for AdventureWorks2019';
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-22T13:21:23.1184550+03:00

- Veritabanında şifrelemeyi aktif edelim.
- Bu adımda TDECertificate ile veritabanına bir şifreleme anahtarı tanımlıyoruz.

```
USE AdventureWorks2019;
GO

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

- Şifreleme işlemini başlatalım.
- 3 = Şifreleme aktif.

```
SELECT db.name, dm.encryption_state, dm.percent_complete, dm.key_algorithm, dm.key_length
FROM sys.dm_database_encryption_keys dm
JOIN sys.databases db
ON dm.database_id = db.database_id;
```

100 %

Results Messages

	name	encryption_state	percent_complete	key_algorithm	key_length
1	tempdb	3	0	AES	256
2	AdventureWorks2019	3	0	AES	256

Adım 4: Basit Bir SQL Injection Zafiyeti Simülasyonu

- Hazırlık: Test için geçici bir tablo oluşturalım.


```
USE AdventureWorks2019;
GO

CREATE TABLE InjectionTest (
    Id INT PRIMARY KEY IDENTITY(1,1),
    Username NVARCHAR(100),
    Password NVARCHAR(100)
);

INSERT INTO InjectionTest (Username, Password) VALUES
('admin', '1234'),
('user1', 'pass1'),
('testuser', 'mypassword');
```

100 %

Messages

(3 rows affected)

Completion time: 2025-04-22T13:24:54.3561236+03:00

- Zayıf ve Güvensiz Sorgu Örneği (Klasik Injection)

```
DECLARE @username NVARCHAR(100) = 'admin' --';
DECLARE @sql NVARCHAR(MAX) =
    'SELECT * FROM InjectionTest WHERE Username = ''' + @username + '''';

EXEC(@sql);
```

100 %

Results Messages

	Id	Username	Password
1	1	admin	1234

- SQL Injection'dan Korunma (Parametrik Sorgu)

```
DECLARE @username NVARCHAR(100) = 'admin' --';
DECLARE @stmt NVARCHAR(MAX) = N'SELECT * FROM InjectionTest WHERE Username = @u';
EXEC sp_executesql @stmt, N'@u NVARCHAR(100)', @u = @username;
```

100 %

Results Messages

	Id	Username	Password
--	----	----------	----------

Adım 5 : SQL Server Audit ile Kullanıcı İzleme (Audit Log)

- Audit nesnesi oluşturalım.

```
USE master;
GO

CREATE SERVER AUDIT Audit_TestUser_Logins
TO FILE (
    FILEPATH = 'C:\SQLAuditLogs\', -- Bu klasör var olmalı!
    MAXSIZE = 5 MB,
    MAX_ROLLOVER_FILES = 10,
    RESERVE_DISK_SPACE = OFF
);
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-22T13:27:46.8049004+03:00

- Audit eylemini (Action Group) tanımlayalım.

```
CREATE SERVER AUDIT SPECIFICATION Audit_Logins_Spec
FOR SERVER AUDIT Audit_TestUser_Logins
ADD (SUCCESSFUL_LOGIN_GROUP),
ADD (FAILED_LOGIN_GROUP);
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-26T12:12:38.1041743+03:00

- Audit'i başlatalım.

```
ALTER SERVER AUDIT Audit_TestUser_Logins WITH (STATE = ON);
```

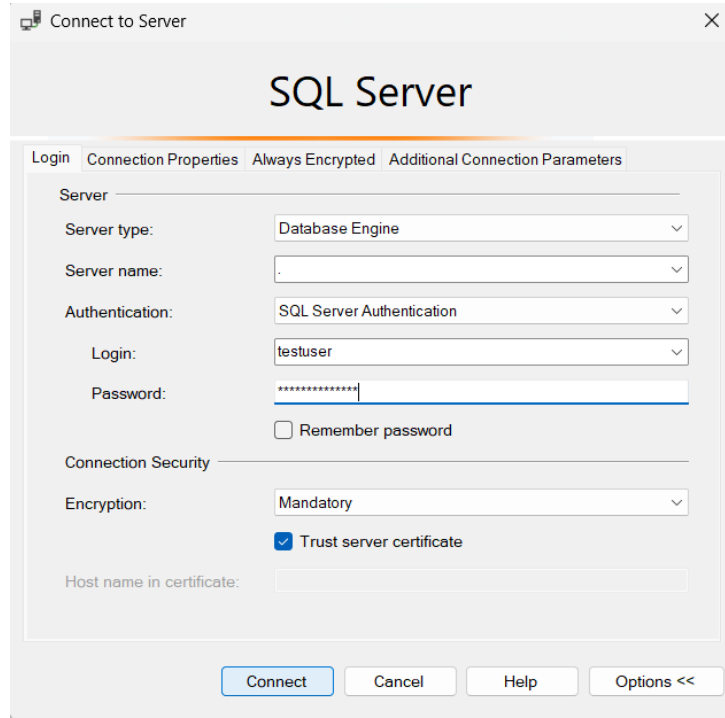
100 %

Messages

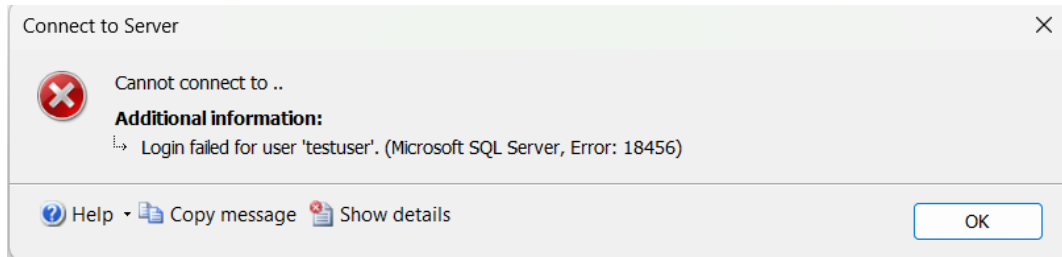
Commands completed successfully.

Completion time: 2025-04-22T13:28:37.9150682+03:00

- Önce yanlış giriş yapalım.

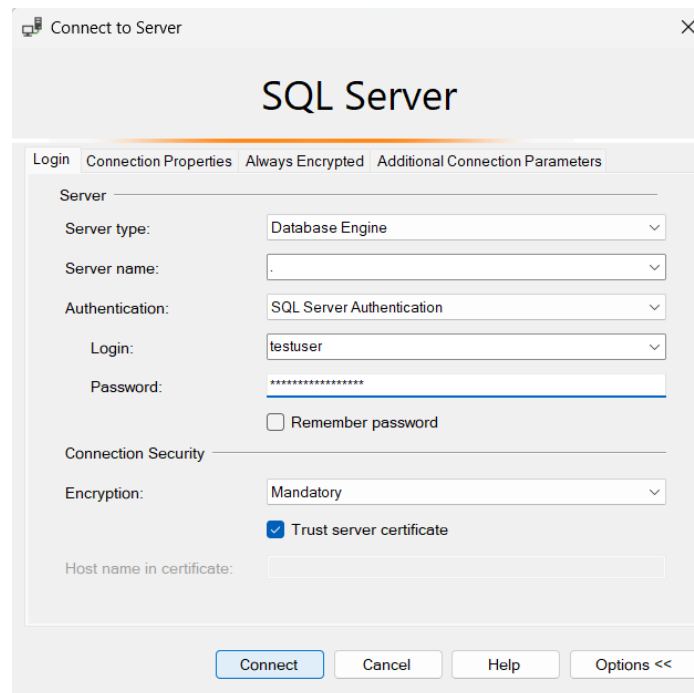


The image shows the 'Connect to Server' dialog box in SQL Server Enterprise Manager. The 'Login' tab is selected. The 'Server' section has 'Server type' set to 'Database Engine', 'Server name' set to '.', 'Authentication' set to 'SQL Server Authentication', 'Login' set to 'testuser', and 'Password' set to a masked password. The 'Remember password' checkbox is unchecked. The 'Connection Security' section has 'Encryption' set to 'Mandatory' and 'Trust server certificate' checked. The 'Host name in certificate' field is empty. At the bottom, there are buttons for 'Connect', 'Cancel', 'Help', and 'Options <<'. The 'Connect' button is highlighted.



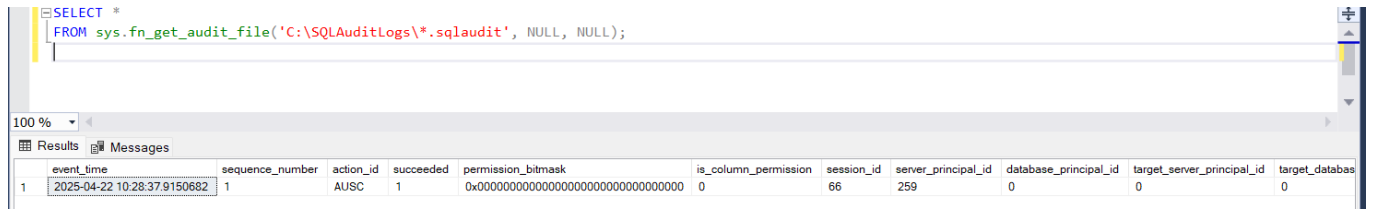
The image shows an error message dialog box titled 'Connect to Server'. It contains a red 'X' icon and the text 'Cannot connect to ..'. Below this, it says 'Additional information:' followed by 'Login failed for user 'testuser'. (Microsoft SQL Server, Error: 18456)'. At the bottom, there are buttons for 'Help', 'Copy message', 'Show details', and 'OK'.

- Şimdi de doğru giriş yapalım.



The image shows the 'Connect to Server' dialog box in SQL Server Enterprise Manager, identical to the one in the first image. The 'Login' tab is selected. The 'Server' section has 'Server type' set to 'Database Engine', 'Server name' set to '.', 'Authentication' set to 'SQL Server Authentication', 'Login' set to 'testuser', and 'Password' set to a masked password. The 'Remember password' checkbox is unchecked. The 'Connection Security' section has 'Encryption' set to 'Mandatory' and 'Trust server certificate' checked. The 'Host name in certificate' field is empty. At the bottom, there are buttons for 'Connect', 'Cancel', 'Help', and 'Options <<'. The 'Connect' button is highlighted.

- Audit loglarını görüntüleyelim.



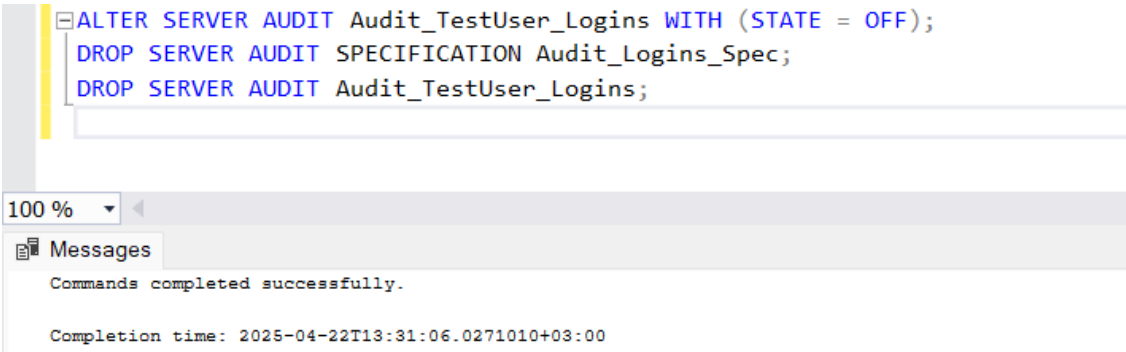
The screenshot shows a SQL query window with the following query:

```
SELECT *
FROM sys.fn_get_audit_file('C:\SQLAuditLogs\*.sqlaudit', NULL, NULL);
```

The Results pane displays a single row of data:

event_time	sequence_number	action_id	succeeded	permission_bitmask	is_column_permission	session_id	server_principal_id	database_principal_id	target_server_principal_id	target_databas
2025-04-22 10:28:37.9150682	1	AUSC	1	0x00000000000000000000000000000000	0	66	259	0	0	0

- Audit'i devre dışı bırakalım.



The screenshot shows a SQL query window with the following commands:

```
ALTER SERVER AUDIT Audit_TestUser_Logins WITH (STATE = OFF);
DROP SERVER AUDIT SPECIFICATION Audit_Logins_Spec;
DROP SERVER AUDIT Audit_TestUser_Logins;
```

The Messages pane shows the following output:

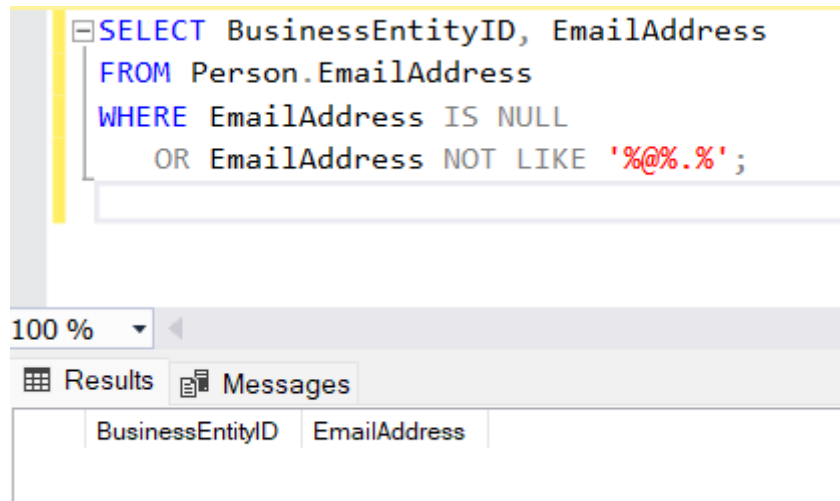
```
Commands completed successfully.

Completion time: 2025-04-22T13:31:06.0271010+03:00
```

PROJE 5: VERİ TEMZİLEME VE ETL SÜREÇLERİ TASARIMI

Adım 1: Hatalı Verileri Tespit Etme

- E-poste verilerinde hatalı veriler var mı kontrol edelim.



The screenshot shows a SQL query window with the following query:

```
SELECT BusinessEntityID, EmailAddress
FROM Person.EmailAddress
WHERE EmailAddress IS NULL
OR EmailAddress NOT LIKE '%@%.%';
```

The Results pane shows the following columns:

BusinessEntityID	EmailAddress
------------------	--------------

- Person.EmailAddress tablosunda yapılan sorguda NULL veya yanlış formatlı e-posta adresi bulunmamıştır. Veriler bu açıdan temiz durumdadır.
- Telefon numaralarındaki formatları kontrol edelim.

SELECT BusinessEntityID, PhoneNumber
FROM Person.PersonPhone
WHERE PhoneNumber NOT LIKE '[0-9][0-9][0-9]-%';

100 %

Results Messages

	BusinessEntityID	PhoneNumber
1	1959	1 (11) 500 555-0110
2	2409	1 (11) 500 555-0110
3	2467	1 (11) 500 555-0110
4	2488	1 (11) 500 555-0110
5	2510	1 (11) 500 555-0110
6	2553	1 (11) 500 555-0110
7	2621	1 (11) 500 555-0110
8	2625	1 (11) 500 555-0110
9	3637	1 (11) 500 555-0110
10	3843	1 (11) 500 555-0110
11	3884	1 (11) 500 555-0110
12	4341	1 (11) 500 555-0110
13	4348	1 (11) 500 555-0110
14	4511	1 (11) 500 555-0110
15	4651	1 (11) 500 555-0110
16	4772	1 (11) 500 555-0110
17	4871	1 (11) 500 555-0110
18	4920	1 (11) 500 555-0110
19	4973	1 (11) 500 555-0110
20	5239	1 (11) 500 555-0110
21	5303	1 (11) 500 555-0110
22	5454	1 (11) 500 555-0110
23	5851	1 (11) 500 555-0110
24	6089	1 (11) 500 555-0110
25	6117	1 (11) 500 555-0110
26	6134	1 (11) 500 555-0110
27	6182	1 (11) 500 555-0110
28	6404	1 (11) 500 555-0110
29	6542	1 (11) 500 555-0110
30	6544	1 (11) 500 555-0110
31	6802	1 (11) 500 555-0110

- Person.PersonPhone tablosunda yer alan telefon numaralarının bir kısmı belirlenen format (XXX-XXX-XXXX) ile uyumlu değildir. Yapılan sorgu sonucu, standart dışı formatta birçok kayıt tespit edilmiştir.

Adım 2: Geçici Tablo Oluştur ve Hatalı Verileri Aktar

- Geçici tablo oluşturalım.

CREATE TABLE #CleanedPhones (
BusinessEntityID INT,
OriginalPhone NVARCHAR(50),
CleanedPhone NVARCHAR(50)
);

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-24T14:33:52.0375503+03:00

- Hatalı verileri aktaralım.

```

INSERT INTO #CleanedPhones (BusinessEntityID, OriginalPhone)
SELECT BusinessEntityID, PhoneNumber
FROM Person.PersonPhone
WHERE PhoneNumber NOT LIKE '[0-9][0-9][0-9]-%';

```

100 %

Messages

(9188 rows affected)

Completion time: 2025-04-24T14:34:09.3275579+03:00

Adım 4: Temiz Telefon Formatı Oluşturma (Transform) (Standart Formata Dönüştürme)

- Temizlenecek sütun ekleyelim.

```

ALTER TABLE #CleanedPhones ADD DigitsOnly NVARCHAR(20);

```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-24T14:36:48.7530346+03:00

- Parantez, boşluk ve tireleri temizleyelim.

```

UPDATE #CleanedPhones
SET DigitsOnly = REPLACE(
    REPLACE(
        REPLACE(
            REPLACE(OriginalPhone, '(', ''),
            ')', ''),
        '-', ''),
    ' ', '');

```

100 %

Messages

(9188 rows affected)

Completion time: 2025-04-24T14:38:07.1011763+03:00

- İlk 10 haneyi alıp XXX-XXX-XXXX formatına çevirelim.

```
UPDATE #CleanedPhones
SET CleanedPhone =
STUFF(
    STUFF(
        SUBSTRING(DigitsOnly, LEN(DigitsOnly)-9, 10),
        4, 0, '-'
    ),
    8, 0, '-'
)
WHERE LEN(DigitsOnly) >= 10;
```

100 %

Messages

(9188 rows affected)

Completion time: 2025-04-24T14:38:47.0325972+03:00

- Sonucu kontrol edelim.

```
SELECT * FROM #CleanedPhones;
```

100 %

Results Messages

	BusinessEntityID	OriginalPhone	CleanedPhone	DigitsOnly
1	1959	1 (11) 500 555-0110	500-555-0110	1115005550110
2	2409	1 (11) 500 555-0110	500-555-0110	1115005550110
3	2467	1 (11) 500 555-0110	500-555-0110	1115005550110
4	2488	1 (11) 500 555-0110	500-555-0110	1115005550110
5	2510	1 (11) 500 555-0110	500-555-0110	1115005550110
6	2553	1 (11) 500 555-0110	500-555-0110	1115005550110
7	2621	1 (11) 500 555-0110	500-555-0110	1115005550110
8	2625	1 (11) 500 555-0110	500-555-0110	1115005550110
9	3637	1 (11) 500 555-0110	500-555-0110	1115005550110
10	3843	1 (11) 500 555-0110	500-555-0110	1115005550110
11	3884	1 (11) 500 555-0110	500-555-0110	1115005550110
12	4341	1 (11) 500 555-0110	500-555-0110	1115005550110
13	4348	1 (11) 500 555-0110	500-555-0110	1115005550110
14	4511	1 (11) 500 555-0110	500-555-0110	1115005550110
15	4651	1 (11) 500 555-0110	500-555-0110	1115005550110
16	4772	1 (11) 500 555-0110	500-555-0110	1115005550110
17	4871	1 (11) 500 555-0110	500-555-0110	1115005550110
18	4920	1 (11) 500 555-0110	500-555-0110	1115005550110
19	4973	1 (11) 500 555-0110	500-555-0110	1115005550110
20	5239	1 (11) 500 555-0110	500-555-0110	1115005550110
21	5303	1 (11) 500 555-0110	500-555-0110	1115005550110
22	5454	1 (11) 500 555-0110	500-555-0110	1115005550110
23	5851	1 (11) 500 555-0110	500-555-0110	1115005550110
24	6089	1 (11) 500 555-0110	500-555-0110	1115005550110
25	6117	1 (11) 500 555-0110	500-555-0110	1115005550110
26	6134	1 (11) 500 555-0110	500-555-0110	1115005550110
27	6182	1 (11) 500 555-0110	500-555-0110	1115005550110
28	6404	1 (11) 500 555-0110	500-555-0110	1115005550110
29	6542	1 (11) 500 555-0110	500-555-0110	1115005550110
30	6544	1 (11) 500 555-0110	500-555-0110	1115005550110
31	6802	1 (11) 500 555-0110	500-555-0110	1115005550110
32	6896	1 (11) 500 555-0110	500-555-0110	1115005550110

Adım 5: Temiz Verileri Yeni Bir Kalıcı Tabloya Yükleme

- Yeni tablo oluşturalım.

```
CREATE TABLE Person.CleanedPersonPhone (
    BusinessEntityID INT,
    CleanedPhone NVARCHAR(50)
);
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-24T14:39:59.0444978+03:00

- Temiz verileri bu tabloya aktaralım.

```
INSERT INTO Person.CleanedPersonPhone (BusinessEntityID, CleanedPhone)
SELECT BusinessEntityID, CleanedPhone
FROM #CleanedPhones
WHERE CleanedPhone IS NOT NULL;
```

100 %

Messages

(9188 rows affected)

Completion time: 2025-04-24T14:40:50.5929792+03:00

Adım 6: Son Kontrol – Yüklenen Veriyi Göster

```
SELECT * FROM Person.CleanedPersonPhone;
```

100 %

Results Messages

	BusinessEntityID	CleanedPhone
1	1959	500-555-0110
2	2409	500-555-0110
3	2467	500-555-0110
4	2488	500-555-0110
5	2510	500-555-0110
6	2553	500-555-0110
7	2621	500-555-0110
8	2625	500-555-0110
9	3637	500-555-0110
10	3843	500-555-0110
11	3884	500-555-0110
12	4341	500-555-0110
13	4348	500-555-0110
14	4511	500-555-0110
15	4651	500-555-0110
16	4772	500-555-0110
17	4871	500-555-0110
18	4920	500-555-0110
19	4973	500-555-0110
20	5239	500-555-0110
21	5303	500-555-0110
22	5454	500-555-0110
23	5851	500-555-0110
24	6089	500-555-0110
25	6117	500-555-0110
26	6134	500-555-0110
27	6182	500-555-0110
28	6404	500-555-0110
29	6542	500-555-0110
30	6544	500-555-0110
31	6802	500-555-0110
32	6896	500-555-0110
33	6897	500-555-0110
34	7070	500-555-0110

Adım 7: Veri Kalitesi Raporu

```
-- Toplam kayıt sayısı
SELECT COUNT(*) AS TotalOriginalRecords FROM Person.PersonPhone;

-- Hatalı telefon numarası içerenler
SELECT COUNT(*) AS InvalidPhoneCount FROM #CleanedPhones;

-- Başarıyla temizlenen kayıtlar
SELECT COUNT(*) AS ValidCleanedPhones FROM #CleanedPhones WHERE CleanedPhone IS NOT NULL;
```

100 %

Results Messages

	TotalOriginalRecords
1	19972

	InvalidPhoneCount
1	9188

	ValidCleanedPhones
1	9188

TotalOriginalRecords: 19972

Person.PersonPhone tablosundaki toplam kayıt sayısı

InvalidPhoneCount: 9188

Geçici tabloya alınan ve üzerinde işlem yapılan telefon kayıtları (örneğin sadece belirli pattern'e sahip olanlar)

ValidCleanedPhones: 9188

Başarıyla temizlenen ve CleanedPhone alanına yazılan kayıt sayısı