ANKARA ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM4522 - Ağ Tabanlı Paralel Dağıtım Sistemleri

Final Proje Dokümanı

Oğuz Han Odabaşı 18290043

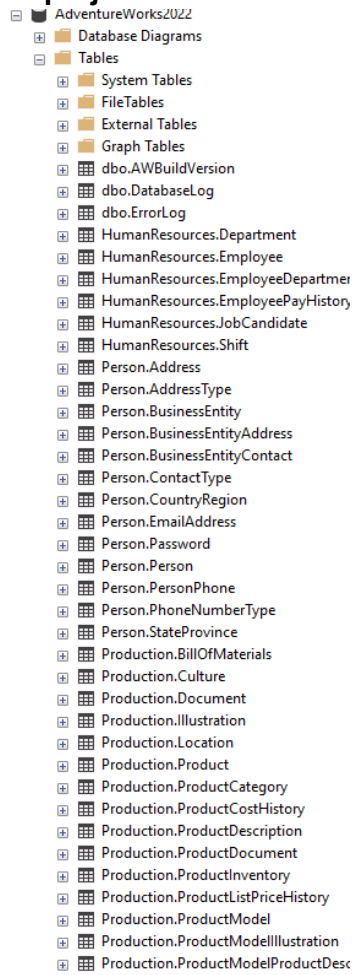
https://github.com/oodabasi/SQL-Rapor

29.05.2025

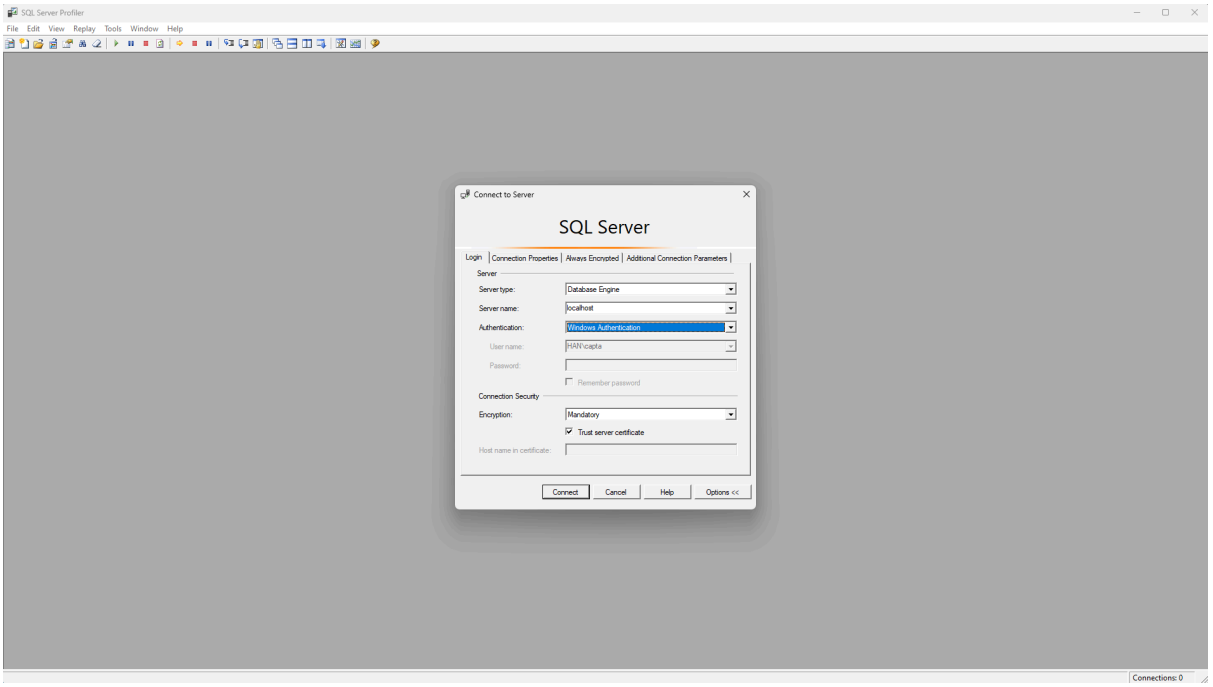
1. Veritabanı Performans Optimizasyonu ve İzleme

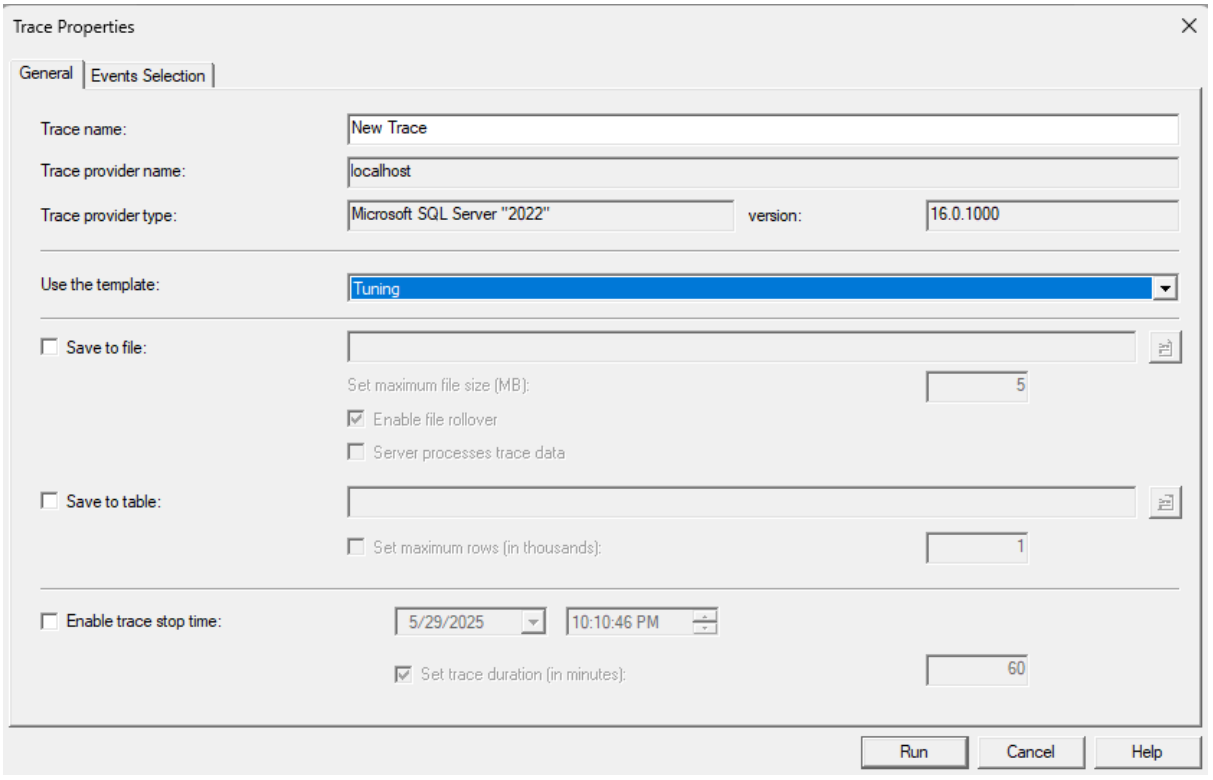
- Büyük bir veritabanı üzerinde performans analizi yaparak optimizasyon tekniklerini uygulamalıdır. Sorgu optimizasyonu, indeks yönetimi, disk alanı yönetimi, ve veri yoğunluğunun yönetimi gibi konular.
    - **Veritabanı İzleme**: SQL Profiler, Dynamic Management Views (DMV) gibi araçlarla sorgu performansını izleme ve hataları tespit etme.
    - **İndeks Yönetimi**: Sorgu hızını artırmak için doğru indekslerin kullanımı, gereksiz indekslerin kaldırılması.
    - **Sorgu İyileştirme**: Uzun süren sorguları analiz etme ve optimize etme.
    - **Veri Yöneticisi Rolleri**: Farklı roller için erişim yönetimi.

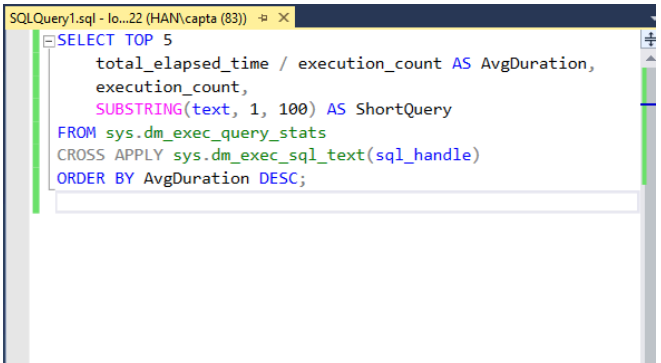
Bu projede AdventureWorks2022 veritabanını kullandım.



Öncelikle SSMS içindeki toollardan SQL Server Profiler ı seçiyoruz.

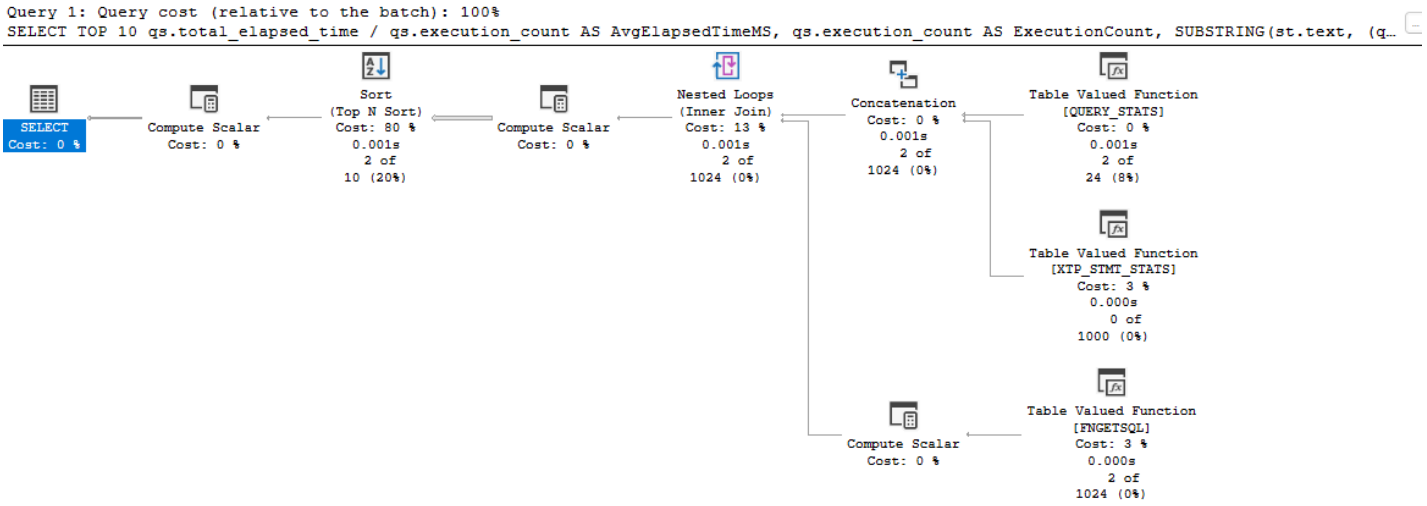




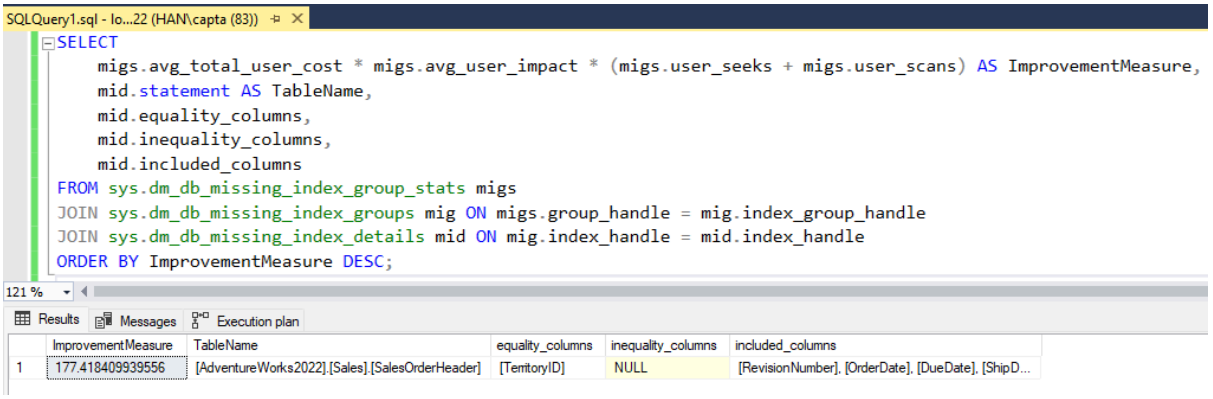


Burada en yavaş çalışan sorgulara bakıyoruz.

Execution plan kısmını açarak (CTRL + M) yavaş çalışan querylerin planlarını görebiliyoruz. Burada direkt yukarıda çalıştırdığım sorguları çalıştırarak outputları paylaşıyorum.



Index Yonetimi:



CREATE NONCLUSTERED INDEX IX_Sales_CustomerID
ON Sales.SalesOrderHeader (CustomerID)
INCLUDE (OrderDate);

```
Commands completed successfully.

Completion time: 2025-05-29T21:40:26.1046426+03:00
```

Mevcut indexleri gostermek:

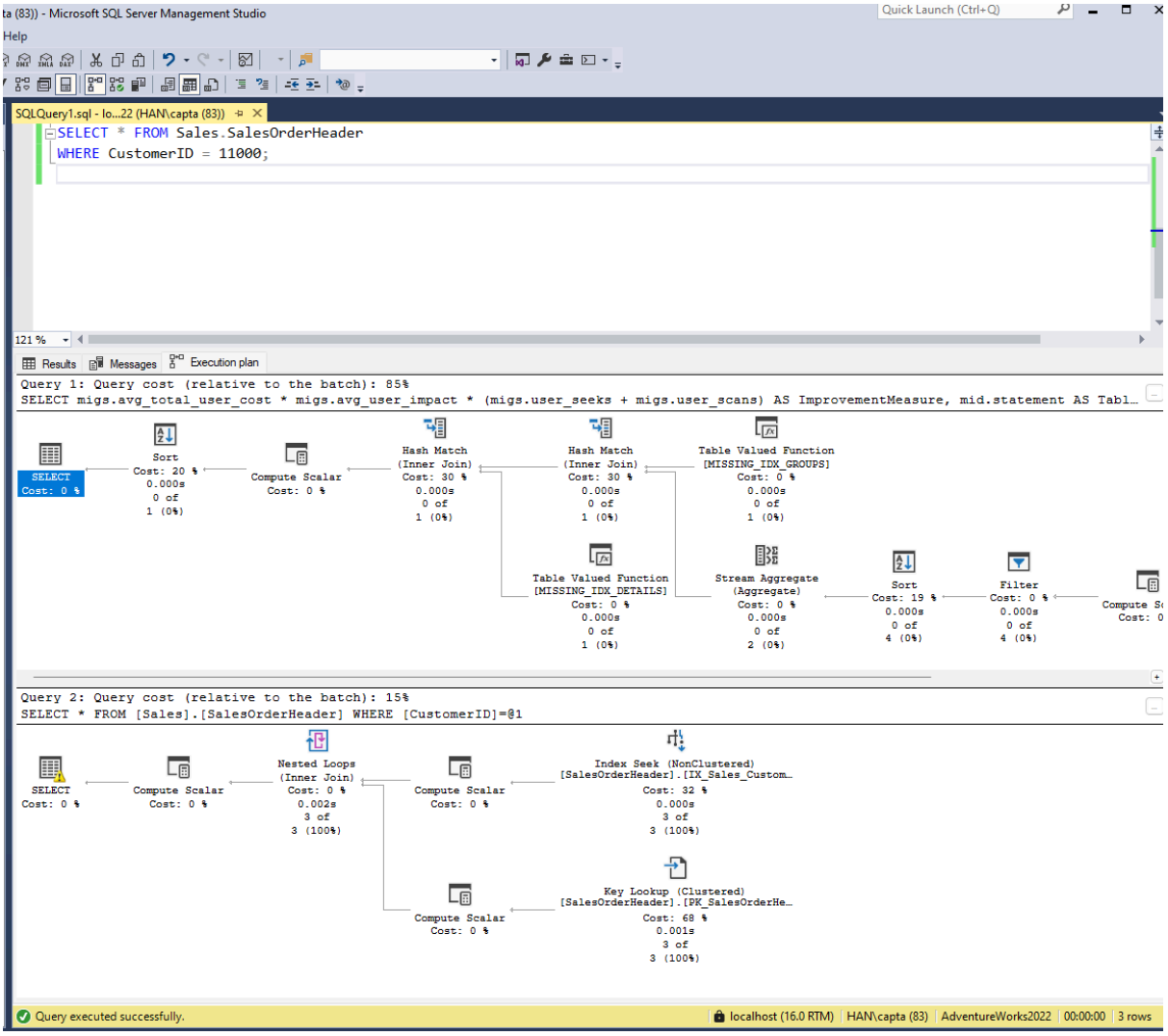
SELECT

t.name AS TableName, i.name AS IndexName, i.type_desc, i.is_disabled, i.is_hypotheticalFROM sys.indexes iJOIN sys.tables t ON i.object_id = t.object_id WHERE i.name IS NOT NULL;

Results | Messages | Execution plan

	TableName	IndexName	type_desc	is_disabled	is_hypothetical
1	SalesTaxRate	PK_SalesTaxRate_SalesTaxRateID	CLUSTERED	0	0
2	SalesTaxRate	AK_SalesTaxRate_StateProvinceID_TaxType	NONCLUSTERED	0	0
3	SalesTaxRate	AK_SalesTaxRate_rowguid	NONCLUSTERED	0	0
4	PersonCreditCard	PK_PersonCreditCard_BusinessEntityID_CreditCardID	CLUSTERED	0	0
5	PersonPhone	PK_PersonPhone_BusinessEntityID_PhoneNumber_Phone...	CLUSTERED	0	0
6	PersonPhone	IX_PersonPhone_PhoneNumber	NONCLUSTERED	0	0
7	SalesTerritory	PK_SalesTerritory_TerritoryID	CLUSTERED	0	0
8	SalesTerritory	AK_SalesTerritory_Name	NONCLUSTERED	0	0
9	SalesTerritory	AK_SalesTerritory_rowguid	NONCLUSTERED	0	0
10	PhoneNumberType	PK_PhoneNumberType_PhoneNumberTypeID	CLUSTERED	0	0

Tablo 170 tane elementten oluştuğu için sadece ilk 10'unu paylastim.



Execution plan kullanarak sorgu sırasında bir indeks eksigi var mi yok mu gorebiliriz.

Veri yonetimi rolleri:

CREATE LOGIN TestUser WITH PASSWORD = 'Test123!';

CREATE USER TestUser FOR LOGIN TestUser;

Burada kullanici oluşturuyoruz.

Rol atama(Sadece okuma yetkisi):

ALTER ROLE db_datareader ADD MEMBER TestUser;

Rol atama(Yazma yetkisi)
ALTER ROLE db_datawriter ADD MEMBER TestUser;
Tum kullanicilari gorme:
*SELECT * FROM sys.database_principals*
WHERE type IN ('S', 'U');

	name	principal_id	type	type_desc	default_schema_name	create_date	modify_date	owning_principal_id	sid	is_fixed_role
1	dbo	1	S	SQL_USER	dbo	2003-04-08 09:10:42.287	2023-05-08 12:26:59.023	NULL	0x01	0
2	guest	2	S	SQL_USER	guest	2003-04-08 09:10:42.317	2003-04-08 09:10:42.317	NULL	0x00	0
3	INFORMATION_SCHEMA	3	S	SQL_USER	NULL	2009-04-13 12:59:11.717	2009-04-13 12:59:11.717	NULL	NULL	0
4	sys	4	S	SQL_USER	NULL	2009-04-13 12:59:11.717	2009-04-13 12:59:11.717	NULL	NULL	0
5	TestUser	5	S	SQL_USER	dbo	2025-05-29 21:50:10.670	2025-05-29 21:50:10.670	NULL	0xAE5F8F21D7520843BC4813FFBB4AB8D5	0

4. Veritabanı Yük Dengeleme ve Dağıtık Veritabanı Yapıları

- Birden fazla veritabanının yönetilmesini, yük dengeleme stratejilerini ve replikasyon teknikleri.
  - **Veritabanı Replikasyonu**: **SQL Server Replication** ile veri çoğaltma ve senkronizasyon sağlama.
  - **Yük Dengeleme**: SQL Server'ı **Always On Availability Groups** veya **Database Mirroring** kullanarak yapılandırma.
  - **Failover Senaryoları**: Yük dengeleme için başarısız bir sunucuya geçiş stratejilerinin uygulanması.

4. Projede karşılaştığım SQL Server Replication u yükleyememe sorunundan dolayı 4. Projeyi yapamıyorum.
Setup wizard ının içinden yüklendiğini söylüyor sitesinde de ancak açtığımda o seçenek malesef çıkmıyor bu yüzden bu projeyi geçip 6. Projeye ilerliyorum.

6. Veritabanı Yükseltme ve Sürüm Yönetimi

- Mevcut bir veritabanını daha yeni bir sürüme yükseltme ve sürüm kontrolü süreçleri.
  - **Veritabanı Yükseltme Planı**: Eski sürümden yeni sürüme geçiş için strateji oluşturma.
  - **Sürüm Yönetimi**: Veritabanı yapısındaki değişikliklerin izlenmesi (örneğin, **DDL Triggers** kullanarak şema değişikliklerini takip etme).
  - **Test ve Geri Dönüş Planı**: Yükseltme sonrası test ve geri dönüş planları.

İlk aşama - Mevcut veritabanının yedeğini alma:
BACKUP DATABASE AdventureWorks2022
TO DISK = 'C:\Backup\AdventureWorks2022.bak';

```
Processed 25496 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.
Processed 2 pages for database 'AdventureWorks2022', file 'AdventureWorks2022_log' on file 1.
BACKUP DATABASE successfully processed 25498 pages in 0.161 seconds (1237.262 MB/sec).

Completion time: 2025-05-29T22:52:36.7929944+03:00
```

İkinci aşama- Yeni sunucuda restore etme:
RESTORE DATABASE AdventureWorks2022
FROM DISK = 'C:\Backup\AdventureWorks2022.bak'
WITH MOVE 'AdventureWorks2022' TO 'D:\SQLData\AdventureWorks2022.mdf',

```
    MOVE 'AdventureWorks2022_log' TO 'D:\SQLLog\AdventureWorks2022.ldf',
    RECOVERY;
  Processed 25496 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.
  Processed 2 pages for database 'AdventureWorks2022', file 'AdventureWorks2022_log' on file 1.
  RESTORE DATABASE successfully processed 25498 pages in 3.687 seconds (54.027 MB/sec).

  Completion time: 2025-05-29T22:55:56.8761613+03:00
```

Eğer SQL sürümleri arasında uyumsuzluk varsa, DMA aracı analiz edip rapor verecektir.

Sürüm yönetimi:
Bu kısımda DDL Triggerlarını kullanacağız.

```
USE AdventureWorks2022;
GO

CREATE TABLE SchemaChangesLog (
    ID INT IDENTITY(1,1),
    EventTime DATETIME,
    EventType NVARCHAR(100),
    ObjectName NVARCHAR(200),
    ObjectType NVARCHAR(50),
    TSQLCommand NVARCHAR(MAX),
    UserName NVARCHAR(200)
);
GO

CREATE TRIGGER trg_LogSchemaChanges
ON DATABASE
FOR DDL_DATABASE_LEVEL_EVENTS
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO SchemaChangesLog (
        EventTime, EventType, ObjectName, ObjectType, TSQLCommand, UserName
    )
    SELECT
        GETDATE(),
        EVENTDATA().value('(/EVENT_INSTANCE/EventType)[1]', 'NVARCHAR(100)'),
        EVENTDATA().value('(/EVENT_INSTANCE/ObjectName)[1]', 'NVARCHAR(200)'),
        EVENTDATA().value('(/EVENT_INSTANCE/ObjectType)[1]', 'NVARCHAR(50)'),
        EVENTDATA().value('(/EVENT_INSTANCE/TSQLCommand)[1]', 'NVARCHAR(MAX)'),
        SYSTEM_USER;
END;
GO
```

Test edelim:

```
ALTER TABLE Person.EmailAddress ADD TestColumn INT;

SELECT * FROM SchemaChangesLog ORDER BY EventTime DESC;
```

	ID	EventTime	EventType	ObjectName	ObjectType	TSQLCommand	UserName
1	1	2025-05-29 22:58:31.317	ALTER_TABLE	EmailAddress	TABLE	ALTER TABLE Person.EmailAddress ADD TestColumn INT	Han\capta

Test ve Geri Dönüş Planı:
Amaç veritabanı yükseltme işleminden sonra tüm sistemin çalıştığından emin olmak.
Ve bir sorun olursa eski haline kolayca geri dönmek.
İlk adımda tablo sayısı ve satır sayısı karşılaştırılır:

Eski ve yeni sistemde çalıştırılır
SELECT COUNT(*) FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE';

	(No column name)
1	72

Eşit çıktığında sorun yok demektir.
Aynı şekilde müşteri sayısı da karşılaştırılır:
SELECT COUNT(*) FROM Sales.Customer;

	(No column name)
1	19820

Geri dönüş planı:
Yükseltme başarısız olursa eğer:
Önceden aldığımız .bak uzantılı yedeği sunucuya geri yükleriz.
use master
RESTORE DATABASE AdventureWorks2022
FROM DISK = 'C:\Backup\AdventureWorks2022.bak'
WITH REPLACE;

```
Processed 25496 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.
Processed 2 pages for database 'AdventureWorks2022', file 'AdventureWorks2022_log' on file 1.
RESTORE DATABASE successfully processed 25498 pages in 3.322 seconds (59.963 MB/sec).

Completion time: 2025-05-29T23:03:33.9573974+03:00
```

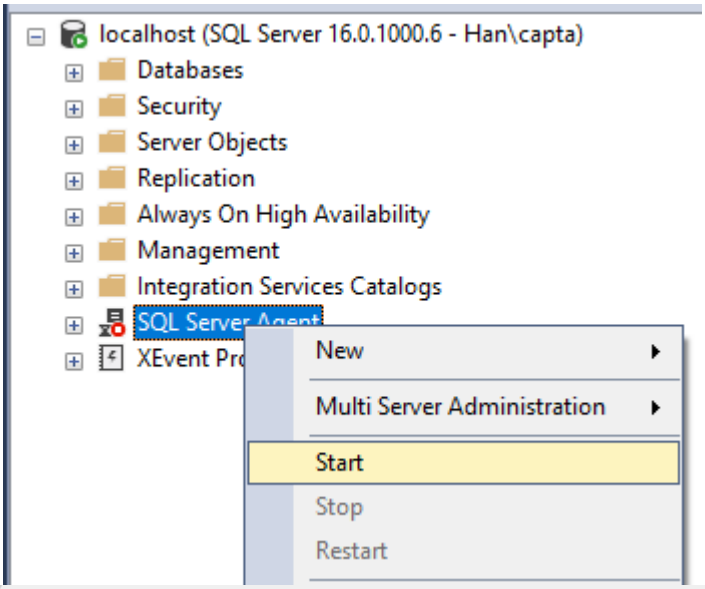
7. Veritabanı Yedekleme ve Otomasyon Çalışması

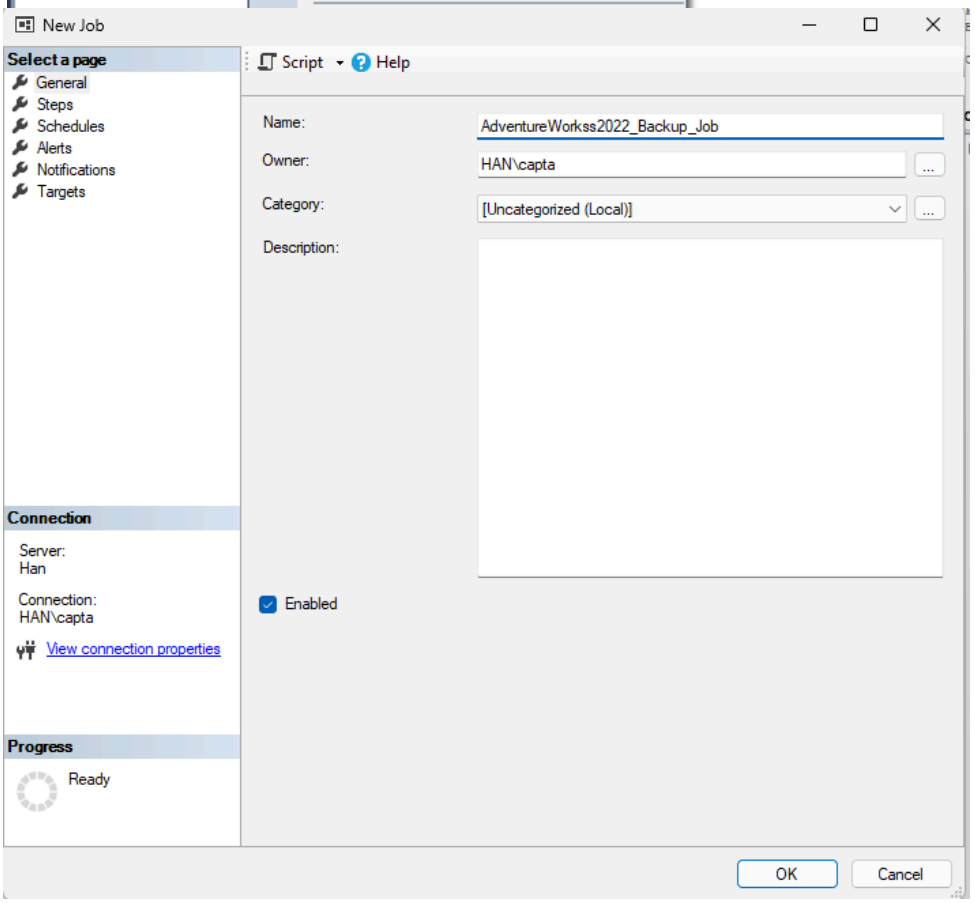
- Veritabanı yedekleme işlemlerini otomatikleştirerek, veritabanı yönetim süreçleri optimize edilir. Ayrıca yedeklerin düzenli olarak alındığını doğrulamak için denetim ve raporlamalar.
  - **SQL Server Agent** kullanarak yedekleme süreçlerini otomatikleştirme.
  - **PowerShell veya T-SQL Scripting** ile yedekleme raporları oluşturma.
  - **Otomatik Yedekleme Uyarıları**: Yedekleme işlemleri başarısız olduğunda yöneticilere bildirim gönderme.

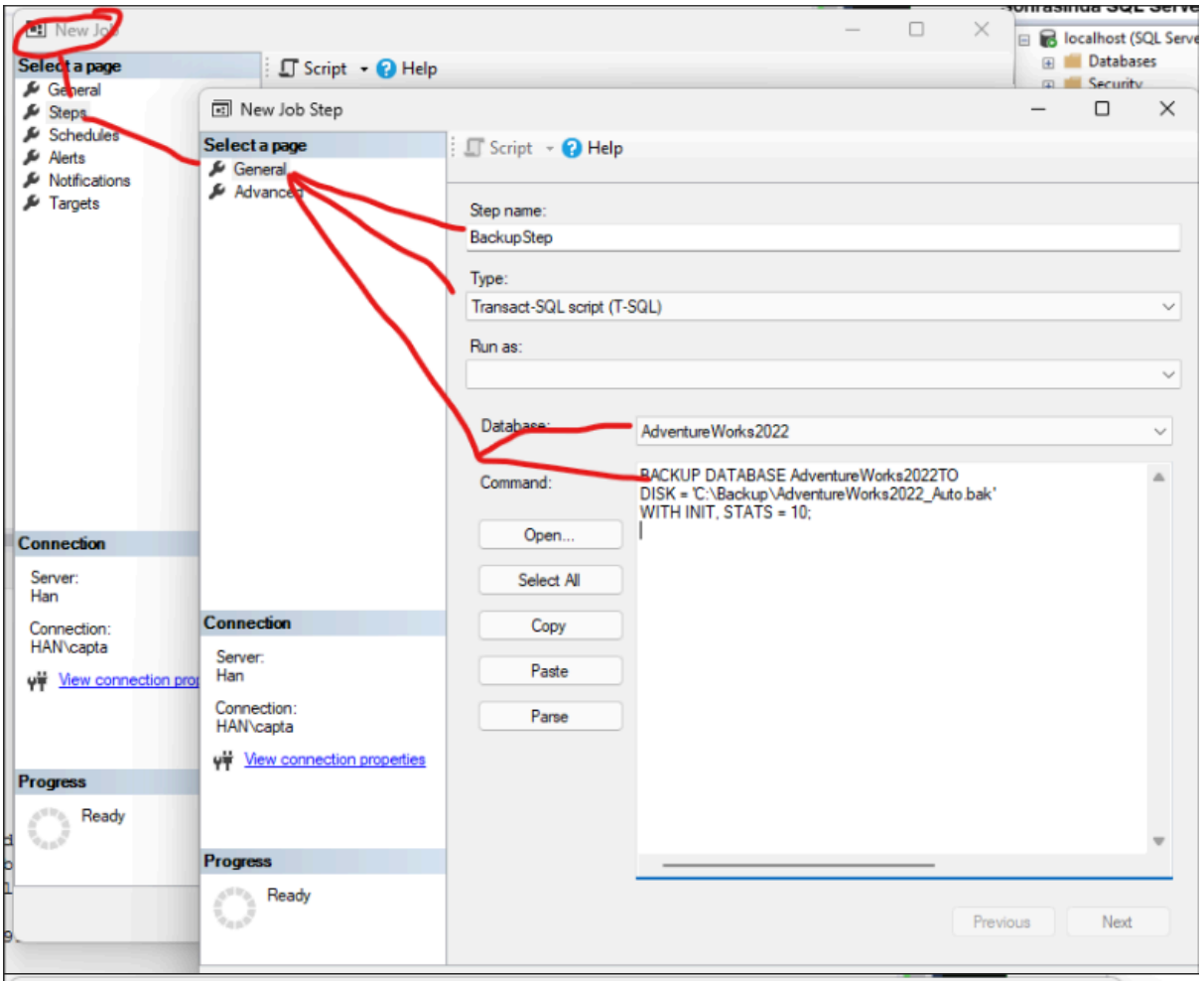
İlk amacımız veritabanının tam yedeğini(Full Backup) almak.
BACKUP DATABASE AdventureWorks2022
TO DISK = 'C:\Backup\AdventureWorks2022_full.bak'
WITH FORMAT, INIT, NAME = 'Full Backup of AdventureWorks2022', STATS = 10;

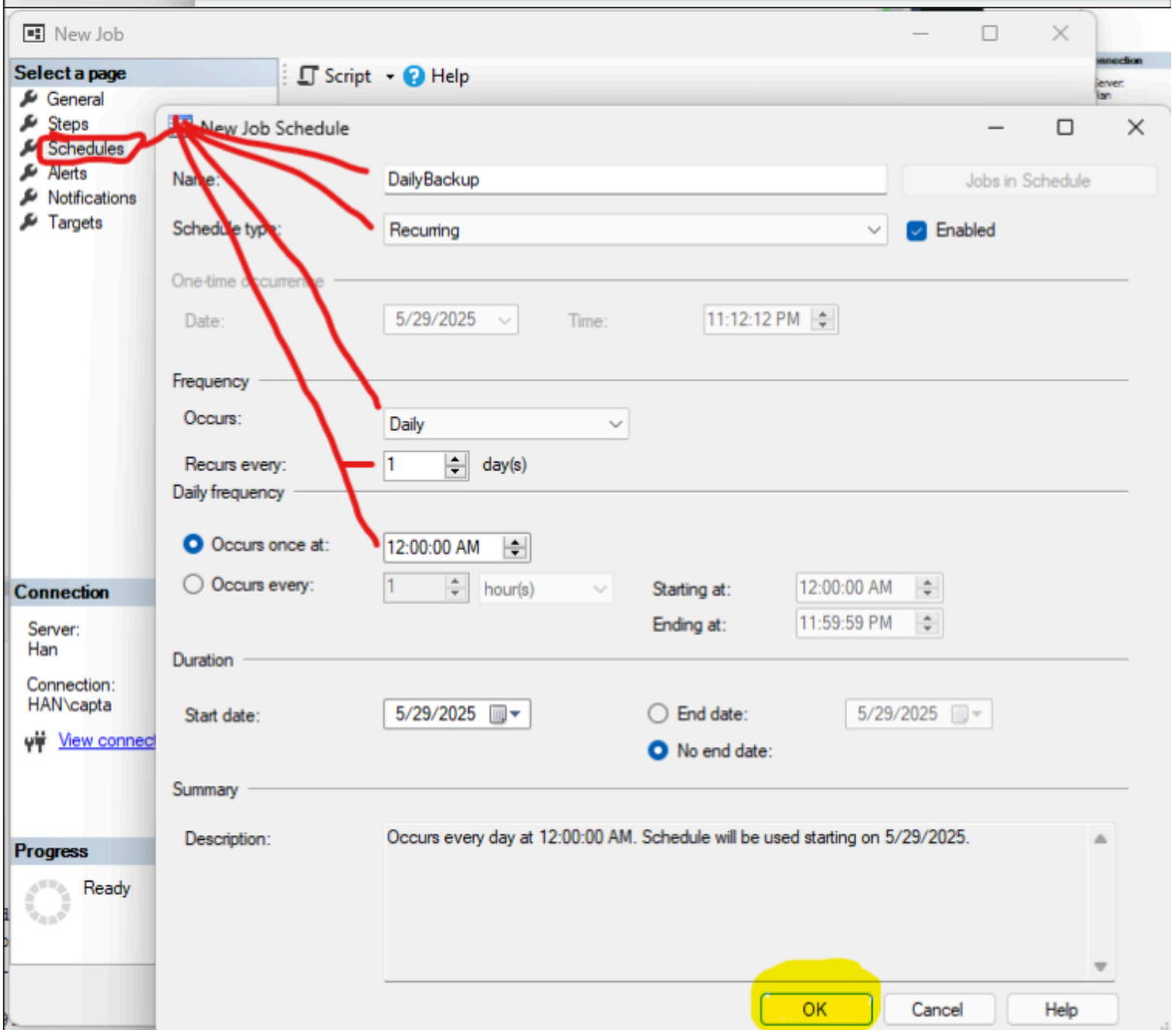
```
10 percent processed.
20 percent processed.
30 percent processed.
40 percent processed.
50 percent processed.
60 percent processed.
70 percent processed.
80 percent processed.
90 percent processed.
100 percent processed.
Processed 25496 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.
Processed 2 pages for database 'AdventureWorks2022', file 'AdventureWorks2022_log' on file 1.
BACKUP DATABASE successfully processed 25498 pages in 3.043 seconds (65.461 MB/sec).

Completion time: 2025-05-29T23:06:55.7413966+03:00
```

Sonrasında SQL Server Agent kullanarak Yedekleme otomasyonunu yapacağız.






New Job
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
New Job Step
Select a page
General
Advanced
Step name:
BackupStep
Type:
Transact-SQL script (T-SQL)
Run as:
Database:
AdventureWorks2022
Command:
BACKUP DATABASE AdventureWorks2022TO
DISK = 'C:\Backup\AdventureWorks2022_Auto.bak'
WITH INIT, STATS = 10;
Open...
Select All
Copy
Paste
Parse
Connection
Server:
Han
Connection:
HAN\capta
View connection properties
Progress
Ready
Previous
Next


New Job
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
New Job Schedule
Name:
DailyBackup
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
5/29/2025
Time:
11:12:12 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:
5/29/2025
End date:
5/29/2025
No end date:
Summary
Description:
Occurs every day at 12:00:00 AM. Schedule will be used starting on 5/29/2025.
OK
Cancel
Help
Connection
Server:
Han
Connection:
HAN\capta
Progress
Ready

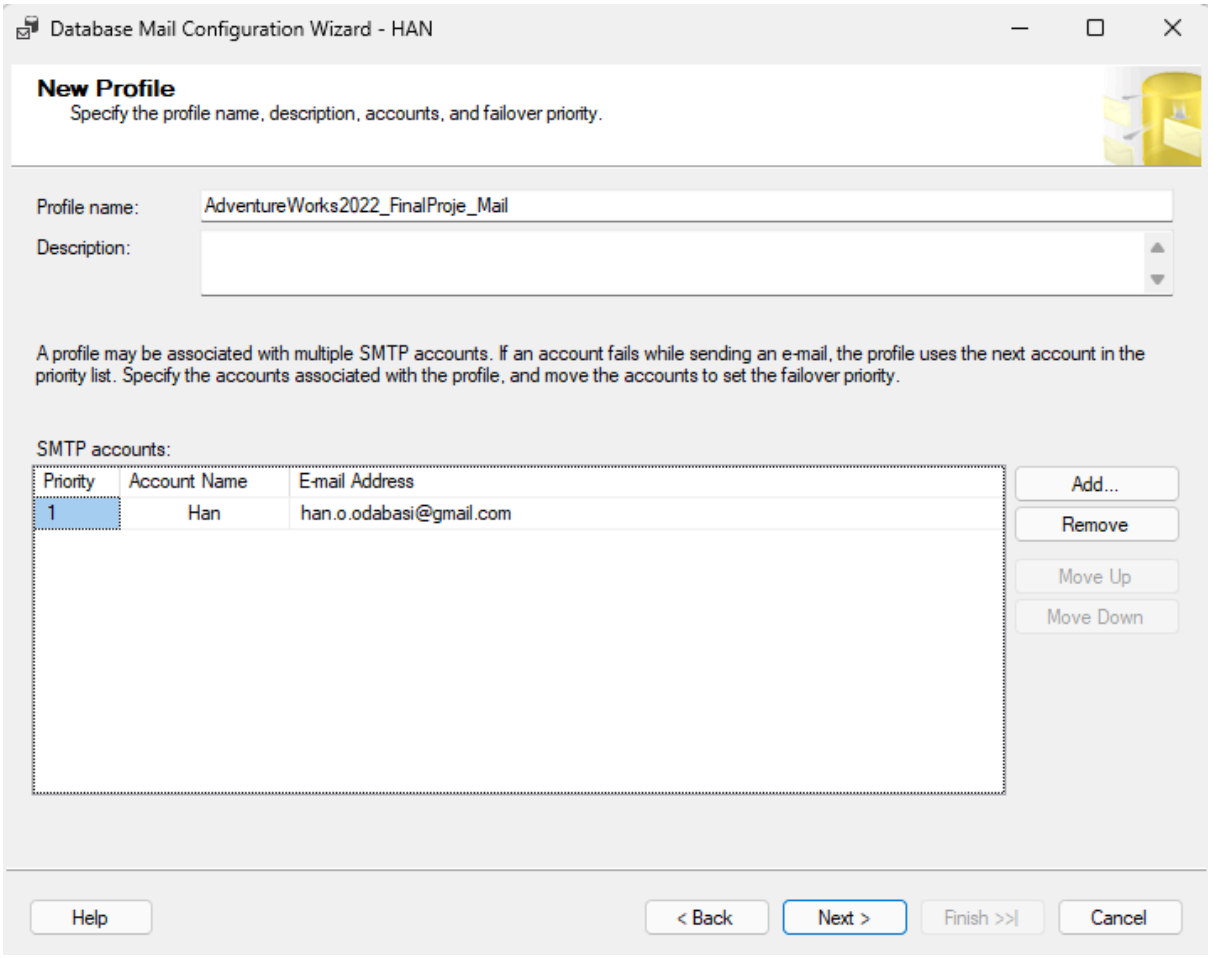
Yedekleme işlemi başarı ile yapıldı mı? Son yedeklemeleri listeleyelim:

```
SELECT
    database_name,
    backup_start_date,
    backup_finish_date,
    backup_size / 1024 / 1024 AS size_mb,
    physical_device_name,
    type =
        CASE
            WHEN type = 'D' THEN 'Full'
            WHEN type = 'I' THEN 'Differential'
            WHEN type = 'L' THEN 'Log'
        END
FROM msdb.dbo.backupset b
JOIN msdb.dbo.backupmediafamily m ON b.media_set_id = m.media_set_id
WHERE database_name = 'AdventureWorks2022'
ORDER BY backup_finish_date DESC;
```

Results | Messages

	database_name	backup_start_date	backup_finish_date	size_mb	physical_device_name
1	AdventureWorks2022	2025-05-29 23:06:52.000	2025-05-29 23:06:55.000	201.08984375000	C:\Backup\AdventureWorks2022_full.bak
2	AdventureWorks2022	2025-05-29 22:54:15.000	2025-05-29 22:54:16.000	201.08593750000	C:\Backup\AdventureWorks2022.bak
3	AdventureWorks2022	2025-05-29 22:52:36.000	2025-05-29 22:52:36.000	201.08984375000	C:\Backup\AdventureWorks2022.bak
4	AdventureWorks2022	2023-05-23 11:56:14.000	2023-05-23 11:56:15.000	200.08593750000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...

Otomatik yedekleme uyarısı ekleme:
Management sekmesinin altında Database Mail a sağ tıklayıp Configure deriz.



Finaldeki projeler burada bitiyor devamında vize için olan raporu tekrar ekliyorum.

2. Veritabanı Yedekleme ve Felaketten Kurtarma Planı

- Bir veritabanının yedekleme ve felaketten kurtarma planlarının tasarlanması. **SQL Server Backup**, **Point-in-time restore**, ve **Database Mirroring** gibi teknikler.
  - **Tam, Artık, Fark Yedeklemeleri**: Yedekleme stratejilerini oluşturma
  - **Zamanlayıcılarla Yedekleme**: Yedekleme işlerini belirli aralıklarla otomatik hale getirme.
  - **Felaketten Kurtarma Senaryoları**: Kaza ile silinen verilerin geri getirilmesi ve kurtarma süreçleri.
  - **Test Yedekleme Senaryoları**: Yedeklerin doğruluğunu test etme.

```
CREATE DATABASE OrnekDB;
GO

USE OrnekDB;
GO

CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY IDENTITY,
    FirstName NVARCHAR(50),
    LastName NVARCHAR(50),
    Email NVARCHAR(100),
    Phone NVARCHAR(20),
    RegistrationDate DATE
);

INSERT INTO Customers (FirstName, LastName, Email, Phone, RegistrationDate)
VALUES
('Berk', 'Dikbas', 'berkdikbas@gmail.com', '05309794343', '2023-01-10'),
('Altug', 'Ozisik', 'altugozisik@gmail.com', '05127478855', '2023-03-22'),
('Aysin', 'Bal', 'aysinbal@gmail.com', '05375472270', '2024-02-14'),
('Arda', 'Turan', 'ardaturan@example.com', '05327651905', '2024-02-14'),
('Fatih', 'Teriml', 'fatihterim@example.com', '05559871234', '2024-02-14'),
('Selcuk', 'Inan', 'selcukinan@example.com', '05559871234', '2024-02-14'),
('Ilhan', 'Mansiz', 'ilhanmansiz@example.com', '05559871234', '2024-02-14'),
('Hakan', 'Balta', 'hakanbalta@example.com', '05559871234', '2024-02-14'),
('Rustu', 'Recber', 'rusturecber@example.com', '05559871234', '2024-02-14'),
('Oguz', 'Odabasi', 'oguzodabasi@example.com', '05559871234', '2024-02-14'),
('Beril', 'sDikbas', 'berildikbas@gmail.com', '05309794242', '2023-01-10');
```

1. Adım: Tam (Full) Yedek Alma

BACKUP DATABASE OrnekDB
TO DISK = 'C:\Backup\OrnekDB_Full.bak'
WITH FORMAT, INIT, NAME = 'Full Backup of OrnekDB';

```
Processed 392 pages for database 'OrnekDB', file 'OrnekDB' on file 1.
Processed 2 pages for database 'OrnekDB', file 'OrnekDB_log' on file 1.
BACKUP DATABASE successfully processed 394 pages in 0.013 seconds (236.478 MB/sec).

Completion time: 2025-04-22T23:54:59.5961482+03:00
```

Full Backup aldıktan sonra kontrol etmek için:

RESTORE VERIFYONLY
FROM DISK = 'C:\Backup\OrnekDB_Full.bak';

```
The backup set on file 1 is valid.

Completion time: 2025-04-22T23:55:25.1618120+03:00
```

2. Adım: Farklı (Differential) Yedek Alma

BACKUP DATABASE OrnekDB
TO DISK = 'C:\Backup\OrnekDB_Diff.bak'
WITH DIFFERENTIAL, NAME = 'Differential Backup of OrnekDB';

```
Processed 120 pages for database 'OrnekDB', file 'OrnekDB' on file 3.
Processed 2 pages for database 'OrnekDB', file 'OrnekDB_log' on file 3.
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 122 pages in 0.011 seconds (86.292 MB/sec).
```

3. Adım: Log (Transaction) Yedeği Alma

BACKUP LOG OrnekDB
TO DISK = 'C:\Backup\OrnekDB_Log.trn'
WITH INIT, NAME = 'Transaction Log Backup of OrnekDB';

```
Processed 11 pages for database 'OrnekDB', file 'OrnekDB_log' on file 1.
BACKUP LOG successfully processed 11 pages in 0.002 seconds (42.968 MB/sec).

Completion time: 2025-04-23T00:03:40.0571640+03:00
```

Veritabanını Full Recovery Mode'a almak gerekebiliyor.

ALTER DATABASE OrnekDB
SET RECOVERY FULL;

4. Adım: Kaza Senaryosu – Veri Kaybı

DELETE FROM Customers;
*SELECT * FROM Customers;*
Tablonun boş olduğunu göreceğiz.

Results Messages

CustomerID	FirstName	LastName	Email	Phone	RegistrationDate

USE master;
ALTER DATABASE OrnekDB SET SINGLE_USER WITH ROLLBACK IMMEDIATE;

5. Adım: Yedekten Geri Yükleme (Restore)
RESTORE DATABASE OrnekDB
FROM DISK = 'C:\Backup\OrnekDB_Full.bak'
WITH REPLACE;

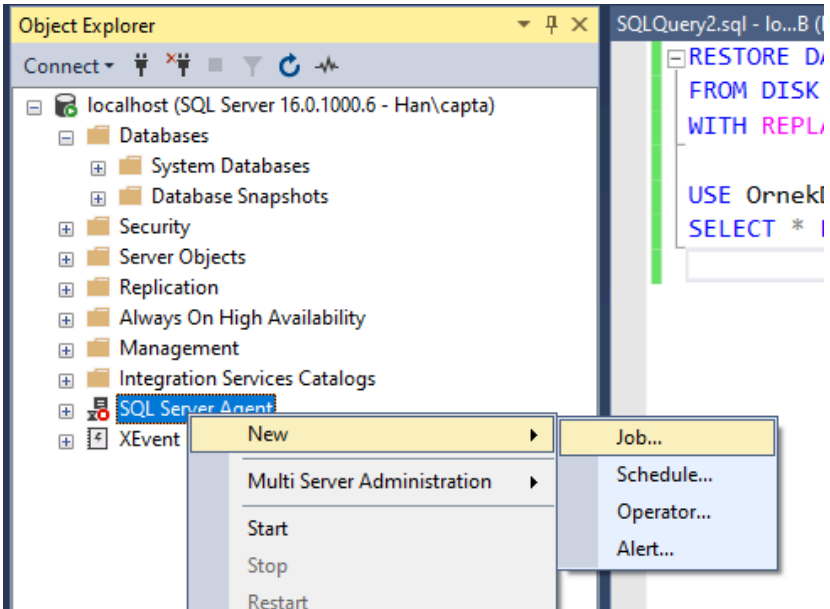
```
Processed 392 pages for database 'OrnekDB', file 'OrnekDB' on file 1.
Processed 2 pages for database 'OrnekDB', file 'OrnekDB_log' on file 1.
RESTORE DATABASE successfully processed 394 pages in 0.007 seconds (439.174 MB/sec).
```

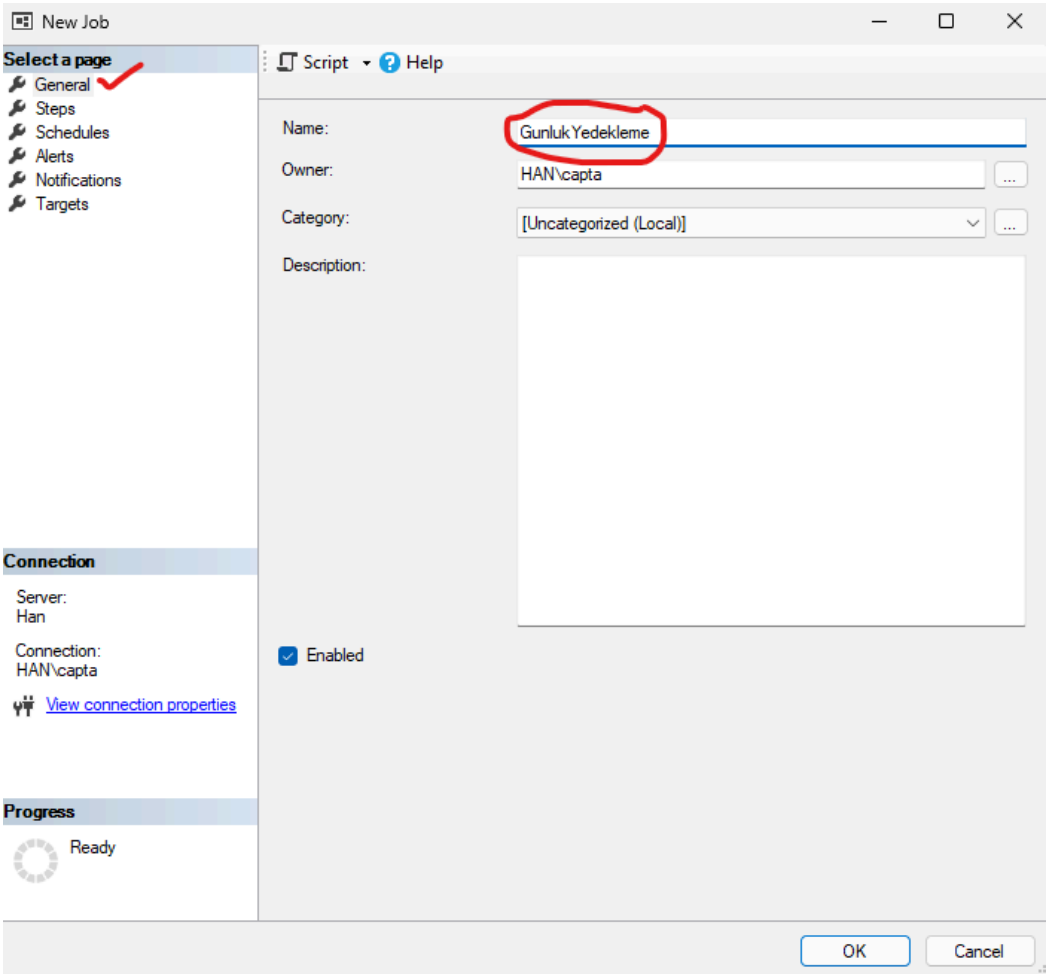
USE OrnekDB;
*SELECT * FROM Customers;*
Tekrar tabloya baktığımızda verilerin geri yüklendiğini görüyoruz.

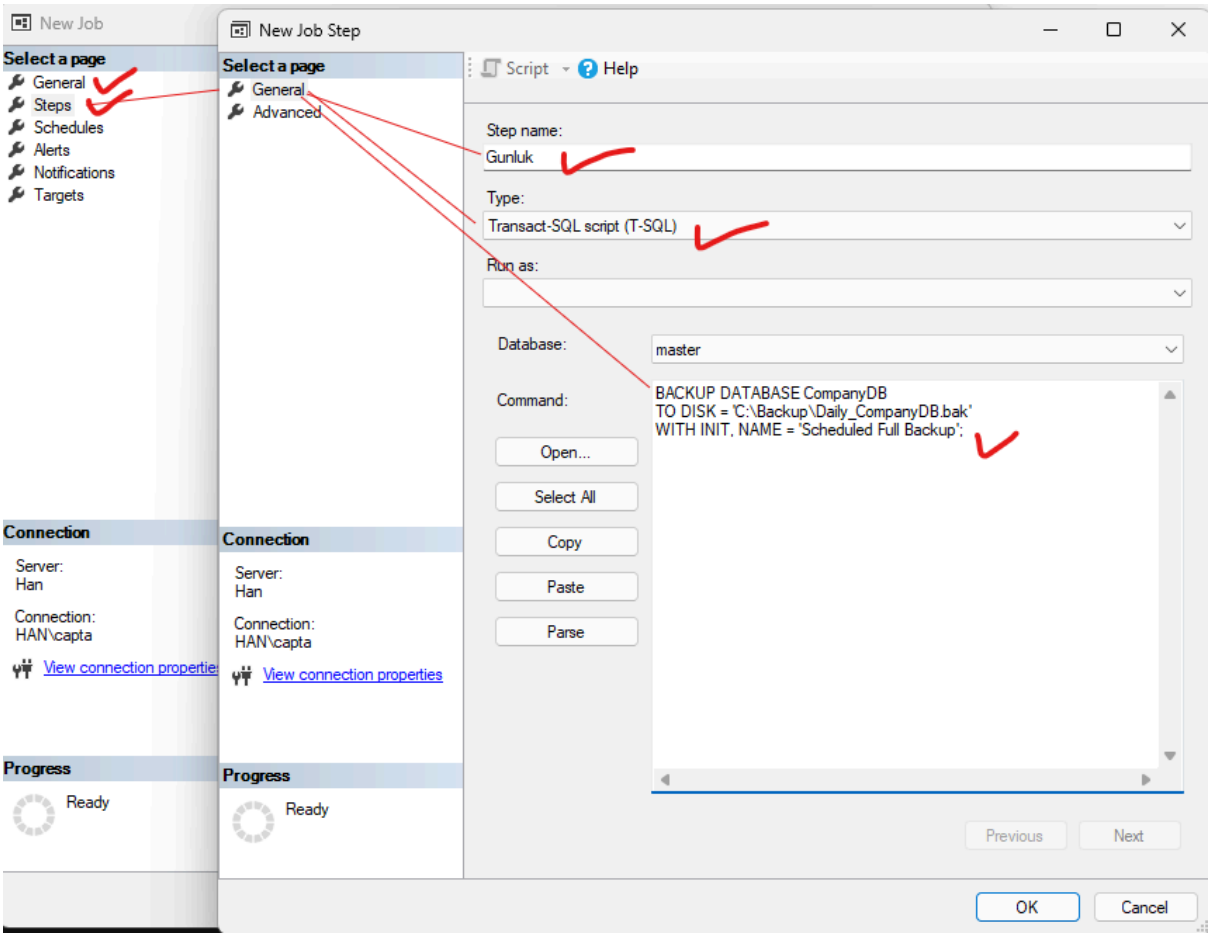
Results Messages

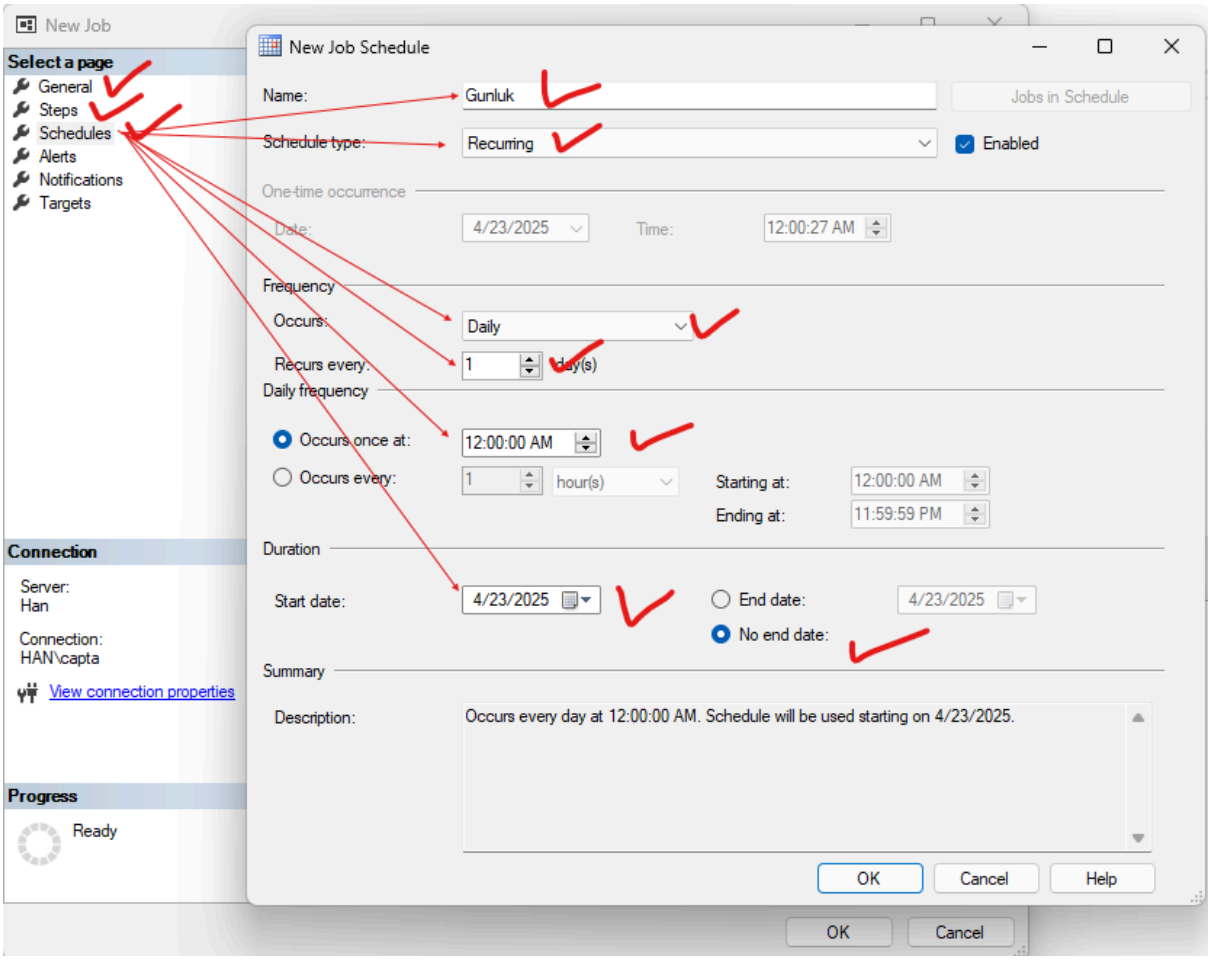
	CustomerID	FirstName	LastName	Email	Phone	RegistrationDate
1	1	Berk	Dikbas	berkdikbas@gmail.com	05309794343	2023-01-10
2	2	Altug	Ozisik	altugozisik@gmail.com	05127478855	2023-03-22
3	3	Aysin	Bal	aysinbal@gmail.com	05375472270	2024-02-14
4	4	Arda	Turan	ardaturan@example.com	05327651905	2024-02-14
5	5	Fatih	Teriml	fatihterim@example.com	05559871234	2024-02-14
6	6	Selcuk	Inan	selcukinan@example.com	05559871234	2024-02-14
7	7	Ilhan	Mansiz	ilhanmansiz@example.com	05559871234	2024-02-14
8	8	Hakan	Balta	hakanbalta@example.com	05559871234	2024-02-14
9	9	Rustu	Recber	rusturecber@example.com	05559871234	2024-02-14
10	10	Oguz	Odabasi	oguzodabasi@example.com	05559871234	2024-02-14
11	11	Beril	sDikbas	berildikbas@gmail.com	05309794242	2023-01-10

Burada otomatik yedekleme için SQL Server Agent adımlarını göstereceğim:




New Job
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
Name:
GunlukYedekleme
Owner:
HAN\capta
Category:
[Uncategorized (Local)]
Description:
Enabled
Connection
Server:
Han
Connection:
HAN\capta
View connection properties
Progress
Ready
OK
Cancel


New Job
New Job Step
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Advanced
Script
Help
Step name:
Gunluk
Type:
Transact-SQL script (T-SQL)
Run as:
Database:
master
Command:
BACKUP DATABASE CompanyDB
TO DISK = 'C:\Backup\Daily_CompanyDB.bak'
WITH INIT, NAME = 'Scheduled Full Backup';
Open...
Select All
Copy
Paste
Parse
Previous
Next
Connection
Server:
Han
Connection:
HAN\capta
View connection properties
Progress
Ready
OK
Cancel



Bu işlemlerden sonra günlük Tam Yedekleme Job'ımız çalışıyor olacak.

---

3. Veritabanı Güvenliği ve Erişim Kontrolü

- Veritabanı güvenliği üzerine odaklanılacak ve özellikle kullanıcı erişimi, veri şifreleme, ve güvenlik duvarı yönetimi gibi konular.
  - **Erişim Yönetimi**: Kullanıcıların verilere erişim yetkilerini yönetmek için **SQL Server Authentication** ve **Windows Authentication** kullanma.
  - **Veri Şifreleme**: Veritabanındaki hassas bilgilerin şifrelenmesi (örneğin, **TDE** - Transparent Data Encryption).
  - **SQL Injection Testleri**: SQL injection saldırılarına karşı veritabanının korunması.
  - **Audit Logları**: Kullanıcı aktivitelerini izlemek için **SQL Server Audit** özelliklerinin kullanımı.

Erişim Yönetimi

SQL Server Authentication Kullanıcısı Oluşturulması

```
CREATE LOGIN test_user WITH PASSWORD = 'StrongPassword123!';
USE OrnekDB;
CREATE USER test_user FOR LOGIN test_user;
EXEC sp_addrolemember 'db_datareader', 'test_user';
EXEC sp_addrolemember 'db_datawriter', 'test_user';

CREATE LOGIN [HAN\capta] FROM WINDOWS;
USE OrnekDB;
CREATE USER [HAN\capta] FOR LOGIN [HAN\capta];
```

	name	type_desc	create_date	is_disabled
1	##MS_PolicyEventProcessingLogin##	SQL_LOGIN	2022-10-08 06:32:02.537	1
2	##MS_PolicyTsqlExecutionLogin##	SQL_LOGIN	2022-10-08 06:32:02.543	1
3	Han\capta	WINDOWS_LOGIN	2025-04-20 17:51:34.737	0
4	LowAccessUser	SQL_LOGIN	2025-04-20 18:25:48.903	0
5	NT AUTHORITY\SYSTEM	WINDOWS_LOGIN	2025-04-20 17:51:34.753	0
6	NT Service\MSSQLSERVER	WINDOWS_LOGIN	2025-04-20 17:51:34.750	0
7	NT SERVICE\SQLSERVERAGENT	WINDOWS_LOGIN	2025-04-20 17:51:35.030	0
8	NT SERVICE\SQLTELEMETRY	WINDOWS_LOGIN	2025-04-20 17:51:35.450	0
9	NT SERVICE\SQLWriter	WINDOWS_LOGIN	2025-04-20 17:51:34.743	0
10	NT SERVICE\Winmgmt	WINDOWS_LOGIN	2025-04-20 17:51:34.747	0
11	sa	SQL_LOGIN	2003-04-08 09:10:35.460	1
12	sql_kullanici	SQL_LOGIN	2025-04-22 23:39:37.260	0
13	test_user	SQL_LOGIN	2025-04-24 17:03:41.110	0

```
USE master;
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'AnotherStrongPassword123!';

CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'TDE Cert';

USE OrnekDB;
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;

ALTER DATABASE OrnekDB SET ENCRYPTION ON;

SELECT * FROM sys.dm_database_encryption_keys;
```

SQLQuery2.sql - lo...B (HAN\capta (63))

```
SELECT * FROM sys.dm_database_encryption_keys;
```

121 %

Results | Messages

	database_id	encryption_state	create_date	regenerate_date	modify_date	set_date	opened_date	key_algorithm	key_leng
1	2	3	2025-04-24 14:14:00.160	2025-04-24 14:14:00.160	2025-04-24 14:14:00.160	1900-01-01 00:00:00.000	2025-04-24 14:14:00.160	AES	256
2	5	3	2025-04-24 14:13:51.723	2025-04-24 14:13:51.723	2025-04-24 14:13:51.723	2025-04-24 14:14:00.153	2025-04-24 14:13:51.723	AES	256

```
DECLARE @name NVARCHAR(50) = 'Oguz';
EXEC sp_executesql
  N'SELECT * FROM Customers WHERE FirstName = @name',
  N'@name NVARCHAR(50)',
  @name = @name;
```

	CustomerID	FirstName	LastName	Email	Phone	RegistrationDate
1	10	Oguz	Odabasi	oguzodabasi@example.com	05559871234	2024-02-14

```
CREATE SERVER AUDIT Audit_Test
TO FILE (FILEPATH = 'C:\\AuditLogs\\');
ALTER SERVER AUDIT Audit_Test WITH (STATE = ON);

USE OrnekDB;
CREATE DATABASE AUDIT SPECIFICATION Audit_CustomersRead
FOR SERVER AUDIT Audit_Test
ADD (SELECT ON OBJECT::[dbo].[Customers] BY [public])
WITH (STATE = ON);
```

SQLQuery2.sql - lo...B (HAN\capta (63))

```
SELECT *
FROM fn_get_audit_file('C:\AuditLogs\*.sqlaudit', DEFAULT, DEFAULT);
```

121 %

Results | Messages

	event_time	sequence_number	action_id	succeeded	permission_bitmask	is_column_permission	session_id	server_principal_id
1	2025-04-20 16:32:46.0558278	1	AUSC	1	0x00000000000000000000000000000000	0	69	259
2	2025-04-20 16:59:29.8904980	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
3	2025-04-22 13:48:45.4820418	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
4	2025-04-22 20:42:17.4132473	1	AUSC	1	0x00000000000000000000000000000000	0	71	259
5	2025-04-22 20:52:13.4852470	1	SL	1	0x00000000000000000000000000000001	1	71	259
6	2025-04-23 08:38:59.0127615	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
7	2025-04-23 08:38:59.0137672	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
8	2025-04-23 20:58:35.0652408	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
9	2025-04-23 20:58:35.0662407	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
10	2025-04-24 08:13:19.3365086	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
11	2025-04-24 08:13:19.3375101	1	AUSC	1	0x00000000000000000000000000000000	0	34	1
12	2025-04-24 14:14:59.0149014	1	AUSC	1	0x00000000000000000000000000000000	0	63	259

5. Veri Temizleme ve ETL Süreçleri Tasarımı

- Büyük veri kümelerinin temizlenmesi ve işlenmesi için **ETL (Extract, Transform, Load)** süreçlerinin oluşturulması. Bu süreç, veri hatalarını tespit etme, veri entegrasyonu ve uyumsuzlukları giderme gibi görevler.
  - **Veri Temizleme**: SQL kullanarak hatalı verilerin (örneğin, eksik, tutarsız, ya da yanlış formatta verilerin) temizlenmesi.
  - **Veri Dönüştürme**: Farklı kaynaklardan gelen verilerin standartlaştırılması ve dönüştürülmesi.
  - **Veri Yükleme**: Verilerin doğru hedef veritabanlarına yüklenmesi.
  - **Veri Kalitesi Raporları**: Veri temizleme ve dönüştürme sürecine dair raporların oluşturulması.

Yanlış Formatta Verilerin Temizlenmesi

```
SELECT *
FROM Customers
WHERE Email NOT LIKE '%_@__%.__%';
```

Bu sorgu, email adresi formatı yanlış olan satırları bulur.

Yanlış Formatlı Verilerin Güncellenmesi

```
UPDATE Customers
SET Email = 'default@example.com'
WHERE Email NOT LIKE '%_@__%.__%';
```

Örnek: Telefon Numarası Formatını Düzenleme

Telefon numaralarının belirli bir formatta (örneğin, 10 haneli) olmasını sağlamak için aşağıdaki SQL sorgusunu kullanabiliriz. Bu, yanlış formatta girilmiş numaraların düzeltilmesini sağlar:

```
UPDATE Customers
SET Phone = CONCAT('0', SUBSTRING(Phone, LEN(Phone) - 9, 10))
WHERE LEN(Phone) > 10;
```

Bu sorgu, telefon numarasının uzunluğu 10'dan fazla olanları bulur ve sadece son 10 haneli kısmı alarak başına '0' ekler.

```
Delete FROM Customers
WHERE Email IS NULL OR Email = '';
```

Bu sorgu, email adresi eksik olan tüm kayıtları listeler.

```
UPDATE Customers
SET RegistrationDate = CONVERT(VARCHAR(10), RegistrationDate, 120);
```

RegistrationDate sütunundaki tarihleri "YYYY-MM-DD" formatına dönüştürür.

```
UPDATE Customers
SET FirstName = UPPER(FirstName), LastName = UPPER(LastName);
```

FirstName ve LastName sütunlarındaki tüm harfleri büyük harfe dönüştürür.

```
USE geciciDB;

IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE
TABLE_NAME = 'Customers')
BEGIN

   CREATE TABLE Customers (
      CustomerID INT PRIMARY KEY IDENTITY(1,1),
      FirstName NVARCHAR(100),
      LastName NVARCHAR(100),
      Email NVARCHAR(255),
      Phone NVARCHAR(15),
      RegistrationDate DATETIME
   );
END;

INSERT INTO Customers (FirstName, LastName, Email, Phone, RegistrationDate)
SELECT FirstName, LastName, Email, Phone, RegistrationDate
FROM OrnekDB.dbo.Customers;
```

Bu sorgu, OrnekDB veritabanındaki Customers tablosundaki tüm verileri geciciDB veritabanındaki aynı isme sahip tabloya aktarır.

Veri Kalitesi Raporları:

```
SELECT *  FROM Customers
WHERE Email IS NULL OR Email = '' OR Phone IS NULL OR Phone = '';

SELECT *  FROM Customers
WHERE LEN(Phone) != 10;
```

Veri temizleme, dönüşüm ve yükleme süreçlerinin otomatikleştirilmesi için SQL Server Job'ları veya benzeri bir araç kullanılabilir. Basit bir örnek olarak, verilerin her gün düzenli olarak temizlenip yüklenmesi için bir zamanlanmış görev oluşturulabilir.