

AĞ TABANLI PARALEL DAĞITIM SİSTEMLERİ

VİZE ÖDEVİ



GITHUB LİNKİ:

<https://github.com/MetehanTRN/AG-TABANLI-PARALEL-DAGITIM-SISTEMLERI>

METEHAN TURAN

20291263

VERİ TABANI GÜVENLİĞİ VE ERİŞİM KONTROLÜ

1. Erişim Yönetimi (SQL Server Authentication):

- 'TestUser', 'AnkaraUni' ve 'Metehan' adlarında SQL Server Authentication kullanıcıları oluşturuldu.

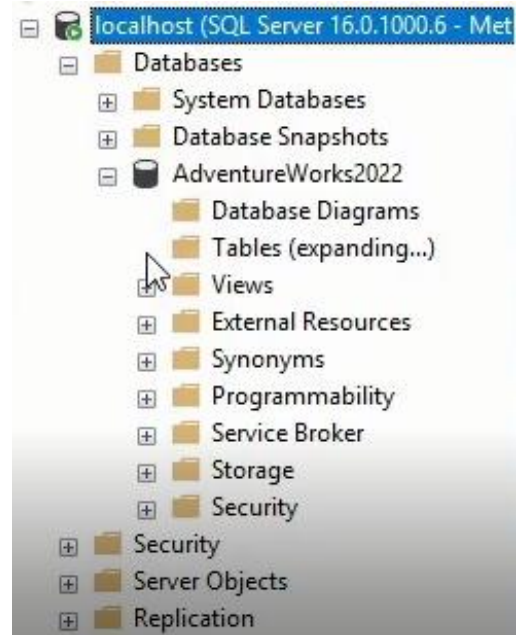
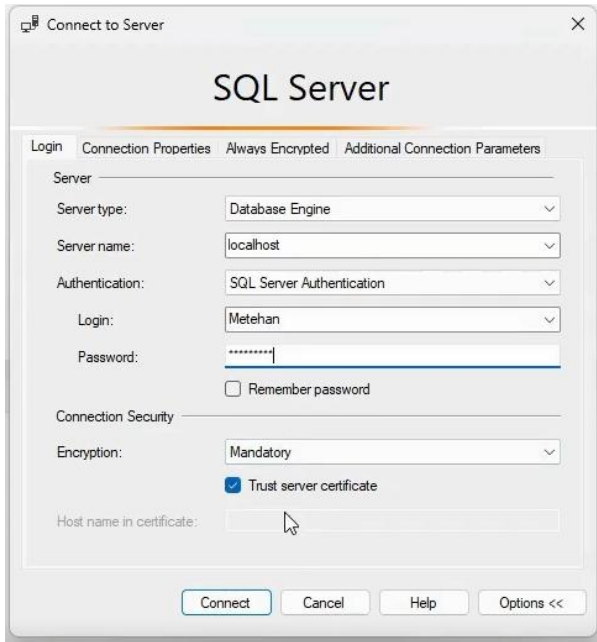
```
SQLQuery2.sql - lo...(MT\metehan (100))* X
CREATE LOGIN Metehan WITH PASSWORD = 'Mete2002.';
```

- Bu kullanıcılar AdventureWorks2022 veritabanında yalnızca okuma yetkisine (db_datareader) sahip olacak şekilde yapılandırıldı.

```
Use AdventureWorks2022;
GO
CREATE USER Metehan FOR LOGIN Metehan;
```

```
EXEC sp_addrolemember 'db_datareader', 'Metehan';
```

- Kullanıcılar ile bağlantı test edildi ve erişim yönetimi başarıyla uygulandı.



2. Transparent Data Encryption (TDE)

- AdventureWorks2022 veritabanında TDE etkinleştirildi ve bir Master Key oluşturuldu.

```
USE master;  
GO  
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'TdeMasterKey123!'  
GO
```

- Ardından bir sertifika oluşturularak veritabanı şifreleme anahtarı tanımlandı.

```
CREATE CERTIFICATE MyTDECert  
WITH SUBJECT = 'AdventureWorks2022 TDE Certificate';  
GO
```

```
USE AdventureWorks2022;  
GO  
CREATE DATABASE ENCRYPTION KEY  
WITH ALGORITHM = AES_256  
ENCRYPTION BY SERVER CERTIFICATE MyTDECert;  
GO
```

- Son adımda ALTER DATABASE ile şifreleme etkinleştirildi.

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

```
SELECT name, is_encrypted
FROM sys.databases
WHERE name = 'AdventureWorks2022';
```

100 %

Results Messages

	name	is_encrypted
1	AdventureWorks2022	1

3. SQL Injection Testleri:

- dbo.Users adında bir test tablosu oluşturuldu ve bazı kullanıcı kayıtları eklendi.

```
CREATE TABLE dbo.Users (
    Id INT PRIMARY KEY IDENTITY,
    Username NVARCHAR(50),
    Password NVARCHAR(50)
);

INSERT INTO dbo.Users (Username, Password)
VALUES ('admin', '1234'), ('metehan', 'ankarauni');
```

- Zararlı kullanıcı giriş senaryosu simüle edildi: 'admin' --' gibi SQL injection örneği ile.

```
DECLARE @username NVARCHAR(50) = 'admin' --';
DECLARE @password NVARCHAR(50) = 'yanlış';

DECLARE @sql NVARCHAR(MAX) =
    'SELECT * FROM dbo.Users WHERE Username = ''' + @username + ''' AND Password = ''' + @password + '''';

EXEC sp_executesql @sql;
```

- sp_executesql ile yapılan doğrudan string birleşimli sorgunun güvensiz olduğu gösterildi.
- Parametrelili sorgularla güvenli bağlantı yöntemi kullanılarak SQL Injection'a karşı koruma sağlandı.

```

DECLARE @username NVARCHAR(50) = 'admin';
DECLARE @password NVARCHAR(50) = '1234';

DECLARE @sql NVARCHAR(MAX) =
    'SELECT * FROM dbo.Users WHERE Username = @u AND Password = @p';

EXEC sp_executesql @sql,
    N'@u NVARCHAR(50), @p NVARCHAR(50)',
    @u = @username, @p = @password;

```

4. SQL Server Audit ile Kullanıcı Takibi:

- “master” veritabanında LoginAudit adında bir audit nesnesi oluşturuldu.

```

USE master;
GO

```

```

CREATE SERVER AUDIT LoginAudit
TO FILE (FILEPATH = 'C:\AuditLogs\');
GO

```

- Audit specification ile başarılı ve başarısız login denemeleri loglandı.

```

CREATE SERVER AUDIT SPECIFICATION LoginAuditSpec
FOR SERVER AUDIT LoginAudit
ADD (SUCCESSFUL_LOGIN_GROUP),
ADD (FAILED_LOGIN_GROUP);
GO

```

- Audit aktif hale getirildi ve giriş denemeleri test edilerek .sqlaudit uzantılı dosyalarda loglandığı doğrulandı.

```

ALTER SERVER AUDIT LoginAudit WITH (STATE = ON);
GO

ALTER SERVER AUDIT SPECIFICATION LoginAuditSpec WITH (STATE = ON);
GO

```

- sys.fn_get_audit_file fonksiyonu kullanılarak loglara erişim sağlandı.

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

	event_time	sequence_number	action_id	succeeded	permission_token	lc_column_permission	session_id	server_principal_id	database_principal_id	target_server_principal_id	target_database_principal_id	object_id	class_type	session_server_principal_name
1	2025-04-24 17:07:02.810098	1	AUSC	1	0-00000000000000000000000000000000	0	108	259	0	0	0	0	A	Microsoft\Account\mstehantur
2	2025-04-24 17:07:14.2996305	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
3	2025-04-24 17:07:14.2996265	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
4	2025-04-24 17:07:14.3013235	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
5	2025-04-24 17:07:14.330454	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
6	2025-04-24 17:07:14.7501773	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
7	2025-04-24 17:07:14.8113816	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
8	2025-04-24 17:07:14.9496422	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
9	2025-04-24 17:07:14.9533618	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur
10	2025-04-24 17:07:14.9562887	1	LGIS	1	0-00000000000000000000000000000000	0	53	259	0	0	0	0	LX	Microsoft\Account\mstehantur

VERİ TEMİZLEME VE ETL SÜREÇLERİ TASARIMI

1. Veri Temizleme:

- Person.Person tablosundaki NULL olan MiddleName değerleri tespit edildi.

```
USE AdventureWorks2022;
GO

SELECT BusinessEntityID, FirstName, MiddleName, LastName
FROM Person.Person
WHERE MiddleName IS NULL;
```

	BusinessEntityID	FirstName	MiddleName	LastName
94	335	Michelle	NULL	Alexander
95	13575	Miguel	NULL	Alexander
96	7293	Miranda	NULL	Alexander
97	7325	Mya	NULL	Alexander
98	6114	Noah	NULL	Alexander
99	19781	Rachel	NULL	Alexander
100	7385	Riley	NULL	Alexander
101	6165	Robert	NULL	Alexander

- ISNULL fonksiyonu ile bu boş alanlar 'NotProvided' olarak değiştirildi.

```
USE AdventureWorks2022;
GO

INSERT INTO Person.CleanedPerson (BusinessEntityID, FirstName, MiddleName, LastName)
SELECT
    BusinessEntityID,
    FirstName,
    ISNULL(MiddleName, 'NotProvided'),
    LastName
FROM Person.Person;
```

- Temizlenen veriler yeni oluşturulan Person.CleanedPerson tablosuna aktarıldı.

```
USE AdventureWorks2022;
GO

CREATE TABLE Person.CleanedPerson (
    BusinessEntityID INT,
    FirstName NVARCHAR(50),
    MiddleName NVARCHAR(50),
    LastName NVARCHAR(50)
);
```

```
USE AdventureWorks2022;
GO

SELECT BusinessEntityID, FirstName, MiddleName, LastName
FROM Person.CleanedPerson;
```

13	16901	Adam	NotProvided	Adams
14	16724	Alex	C	Adams
15	10263	Alexandra	J	Adams
16	10312	Allison	L	Adams
17	10274	Amanda	P	Adams
18	10292	Amber	NotProvided	Adams
19	10314	Andrea	M	Adams
20	16699	Angel	NotProvided	Adams
21	10299	Bailey	NotProvided	Adams
22	1770	Ben	NotProvided	Adams
23	4194	Blake	L	Adams
24	305	Carla	J.	Adams
25	16691	Carlos	NotProvided	Adams

2. Veri Dönüştürme:

- Temizlenen veriler UPPER fonksiyonu ile büyük harfe dönüştürüldü.

```
USE AdventureWorks2022;
GO

INSERT INTO Person.TransformedPerson (BusinessEntityID, FullName)
SELECT
    BusinessEntityID,
    UPPER(FirstName + ' ' + MiddleName + ' ' + LastName)
FROM Person.CleanedPerson;
```

- Ad, ikinci ad ve soyad birleştirilerek FullName alanı oluşturuldu.
- Bu veriler Person.TransformedPerson adlı yeni tabloya kaydedildi.

```
USE AdventureWorks2022;
GO

CREATE TABLE Person.TransformedPerson (
    BusinessEntityID INT,
    FullName NVARCHAR(150)
);
```



```
USE AdventureWorks2022;
GO

SELECT * FROM Person.TransformedPerson;
```

25	16691	CARLOS NOTPROVIDED ADAMS
26	4891	CHARLES R ADAMS
27	10251	CHLOE A ADAMS
28	16872	CONNOR NOTPROVIDED ADAMS
29	10293	COURTNEY C ADAMS
30	4503	DALTON NOTPROVIDED ADAMS
31	4970	DEVIN NOTPROVIDED ADAMS
32	5055	EDUARDO A ADAMS
33	3731	EDWARD NOTPROVIDED ADAMS
34	16858	ELIJAH L ADAMS

3. Veri Yükleme:

- Veriler nihai hedef tablo olan Person.FinalPerson tablosuna yüklendi.

```
USE AdventureWorks2022;
GO

CREATE TABLE Person.FinalPerson (
    BusinessEntityID INT PRIMARY KEY,
    FullName NVARCHAR(150)
);

USE AdventureWorks2022;
GO

INSERT INTO Person.FinalPerson (BusinessEntityID, FullName)
SELECT BusinessEntityID, FullName
FROM Person.TransformedPerson;
```

- Final tabloya yükleme işleminde BusinessEntityID ve dönüştürülmüş FullName kullanıldı.

```
USE AdventureWorks2022;
GO

SELECT * FROM Person.FinalPerson;
```

BusinessEntityID	FullName
1	KEN J SÁNCHEZ
2	TERRI LEE DUFFY
3	ROBERTO NOTPROVIDED TAMBURELLO
4	ROB NOTPROVIDED WALTERS
5	GAIL A ERICKSON
6	JOSSEF H GOLDBERG

4. Veri Kalitesi Raporu:

- Temizlenen veri sayısı hesabı yapıldı.

```
USE AdventureWorks2022;
GO

SELECT COUNT(*) AS NotProvidedCount
FROM Person.CleanedPerson
WHERE MiddleName = 'NotProvided';
```

NotProvidedCount
1 8499

- Dönüştürülen kayıt sayısı hesaplandı.

```
USE AdventureWorks2022;
GO

SELECT COUNT(*) AS TransformedCount
FROM Person.TransformedPerson;
```

TransformedCount
1 19972

- Final tabloya yüklenen kayıt sayısı hesaplandı.

```
USE AdventureWorks2022;  
GO  
  
SELECT COUNT(*) AS FinalLoadCount  
FROM Person.FinalPerson;
```

	FinalLoadCount
1	19972

- Örnek kontrol için ilk 5 kayıt ekrana getirildi.

```
USE AdventureWorks2022;  
GO  
  
SELECT TOP 5 * FROM Person.FinalPerson;
```

Results		Messages
BusinessEntityID	FullName	
1	KEN J SÁNCHEZ	
2	TERRI LEE DUFFY	
3	ROBERTO NOTPROVIDED TAMBURELLO	
4	ROB NOTPROVIDED WALTERS	
5	GAIL A ERICKSON	

VERİTABANI YEDEKLEME VE FELAKETTEN KURTARMA PLANI

1. Veritabanı Yedekleme:

- Tam Yedekleme yapıldı.

```
BACKUP DATABASE AdventureWorks2022  
TO DISK = 'C:\Backup\AdventureWorks\FullBackup.bak'  
WITH INIT, FORMAT;
```

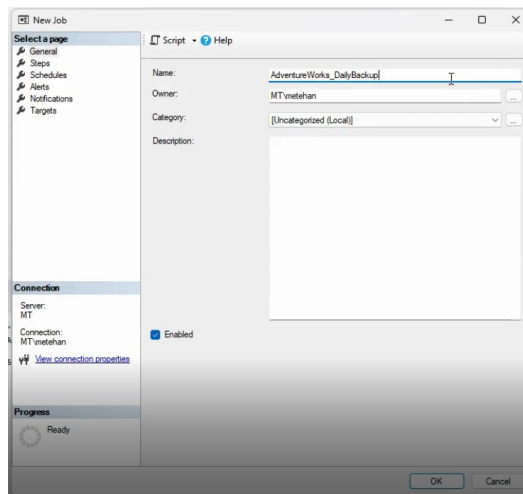
- Artık Yedekleme yapıldı.

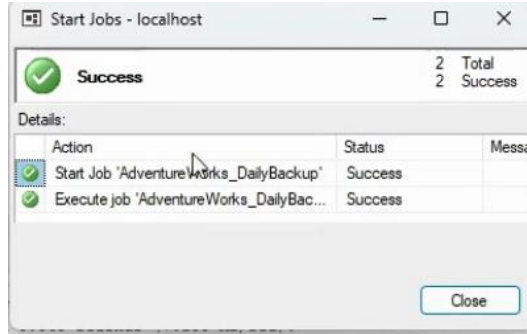
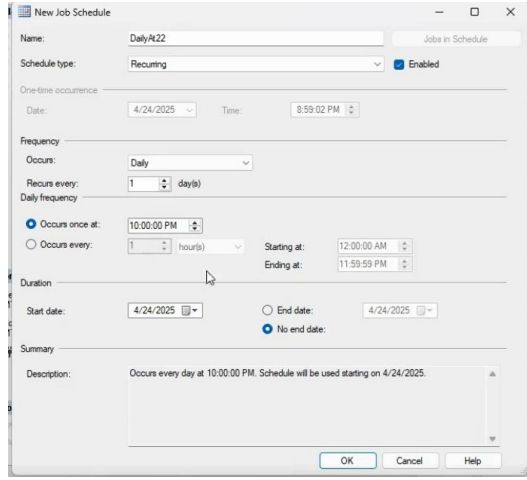
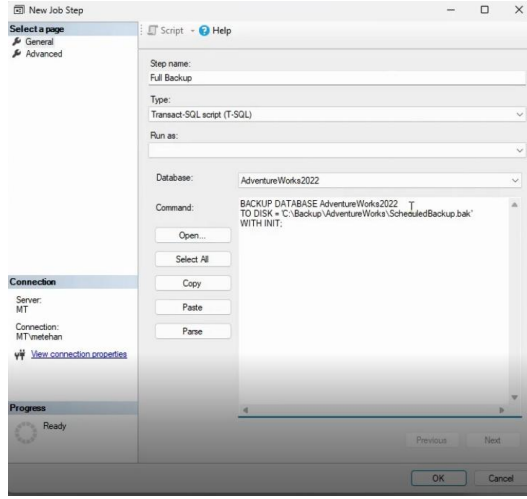
```
BACKUP DATABASE AdventureWorks2022  
TO DISK = 'C:\Backup\AdventureWorks\DiffBackup.bak'  
WITH DIFFERENTIAL;
```

- Fark Yedeklemesi yapıldı.

```
ALTER DATABASE AdventureWorks2022 SET RECOVERY FULL;  
GO  
  
BACKUP LOG AdventureWorks2022  
TO DISK = 'C:\Backup\AdventureWorks\LogBackup.trn';
```

- Zamanlayıcılarla Yedekleme yapıldı.





2. Senaryo: Person.CleanedPerson Tablosunun Kurtarılması:

- Bu senaryoda, AdventureWorks2022 veritabanındaki Person.CleanedPerson tablosu yanlışlıkla silinmiş olarak kabul edilmiştir.
- Bu durumda daha önceden alınmış olan tam veritabanı yedeği (Full Backup) kullanılarak tablo geri getirilmiştir

3. Tablonun Yanlışlıkla Silinmesi:

```
DROP TABLE Person.CleanedPerson;
```

4. Veritabanını Kurtarmaya Hazırlık

```
USE master;  
GO  
  
ALTER DATABASE AdventureWorks2022 SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
```

5. Yedekten Veritabanını Geri Yükleme:

```
RESTORE DATABASE AdventureWorks2022  
FROM DISK = 'C:\Backup\AdventureWorks\FullBackup.bak'  
WITH REPLACE;
```

6. Veritabanını Tekrar Çok Kullanıcı Hale Getirme:

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

7. Geri Yüklenen Tablonun Kontrolü:

```
USE AdventureWorks2022;  
GO  
  
SELECT TOP 5 * FROM Person.CleanedPerson;
```

%

Results Messages

BusinessEntityID	FirstName	MiddleName	LastName
285	Syed	E	Abbas
293	Catherine	R.	Abel
295	Kim	Not Provided	Abercrombie
2170	Kim	Not Provided	Abercrombie
38	Kim	B	Abercrombie