

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4522 - Ağ Tabanlı Paralel Dağıtım Sistemleri

Vize Proje Dokümanı

Oğuz Han Odabaşı 18290043

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

25.04.2025

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', 'Ozsisik', '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).
```

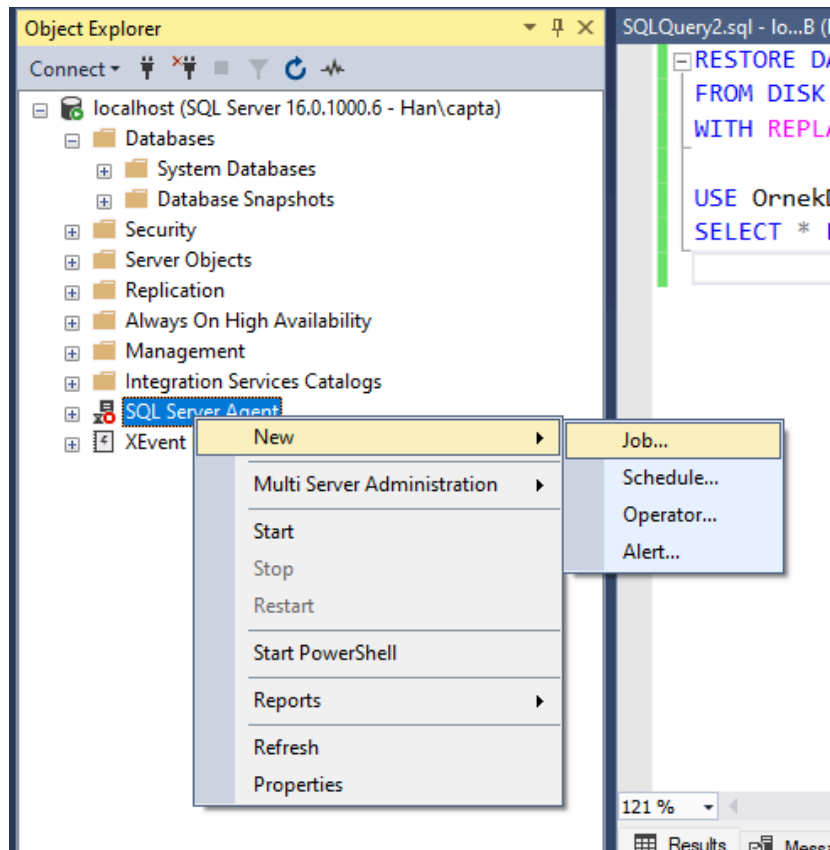
USE OrnekDB;

SELECT * FROM Customers;

Tekrar tabloya baktığımızda verilerin geri yüklendiğini görüyoruz.

	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: Gunluk Yedekleme

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

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

OK Cancel

New Job Schedule

Name: Gunluk ✓

Schedule type: Recurring ✓

Enabled: ☒

One-time occurrence

Date: 4/23/2025 Time: 12:00:27 AM

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

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 şifrlenmesi (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.

Eriřim Yönetimi

SQL Server Authentication Kullanıcısı Oluřturulması

```
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_length
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.