

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM 4522

Ağ Tabanlı Paralel Dağıtım Sistemleri

Şahin Berk Çelik

Veri tabanı Güvenliği ve Erişim Kontrolü 21290630

Github:berk-celik

Ağ Tabanlı Paralel Dağıtım Sistemleri dersinde kullanmak üzere 2024 yılı boyunca orta ölçekli bir organizasyon için tedarik işlemlerini simüle eden bir sentetik veri seti kullandım. Bu veri seti, çeşitli tedarikçilerden ve alıcılardan birden fazla kategorideki (elektronik, mobilya, kırtasiye vb.) satın alımları içerir. 9 sütun, 500 satırdan oluşmaktadır. Küçük bir bölümü aşağıdaki gibidir.

	TransactionID	ItemName	Category	Quantity	UnitPrice	TotalCost	PurchaseDate	Supplier	Buyer
1	TXN001	Desk Chair	Furniture	10	113.15	1131.5	2024-04-19 00:00:00.000	TechMart Inc.	Kelly Joseph
2	TXN002	Stapler	Office Supplies	16	12.62	201.92	2024-07-06 00:00:00.000	CloudSoft Corp.	Kelly Joseph
3	TXN003	Annual Software License	Software	1	5649.34	5649.34	2024-09-10 00:00:00.000	TechMart Inc.	Kelly Joseph
4	TXN004	Notepad	Stationery	13	2.92	37.96	2024-01-21 00:00:00.000	FumiWorks Ltd.	Luis Holland
5	TXN005	Notepad	Stationery	19	1.39	26.41	2024-02-03 00:00:00.000	TechMart Inc.	Cynthia Jenkins
6	TXN006	Printer	Electronics	19	150.94	2867.86	2024-11-28 00:00:00.000	FumiWorks Ltd.	Stephanie Bennett
7	TXN007	Notepad	Stationery	8	2.73	21.84	2024-02-19 00:00:00.000	OfficeSupplies Co.	Todd James
8	TXN008	Notepad	Stationery	4	2.42	9.68	2024-09-11 00:00:00.000	FumiWorks Ltd.	Aaron Hopkins
9	TXN009	Printer Ink	Stationery	13	11.89	154.57	2024-04-12 00:00:00.000	FumiWorks Ltd.	Kevin Adams
10	TXN010	Whiteboard	Furniture	19	100.82	1915.58	2024-12-14 00:00:00.000	FumiWorks Ltd.	Aaron Hopkins

Bu veri setini burada bulabilirsiniz:

<https://www.kaggle.com/datasets/shahriarkabir/company-purchasing-dataset>

1. Veritabanı Yedekleme ve Felaketten Kurtarma Planı

Bu bölümde bir veritabanının yedekleme ve felaketten kurtarma planlarının tasarlanması anlatılacaktır. Sırasıyla Tam, Artımlı, Fark yedeklemeleri, Zamanlayıcılarla yedekleme, Felaketten kurtarma senaryoları ve Test yedekleme senaryoları anlatılacaktır.

1.1. Tam Yedekleme

Burada veritabanının tamamı yedeklenir, komutu şu şekildedir.

```
BACKUP DATABASE BLM4522
TO DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\BLM4522.bak';
```

1.2. Fark Yedekleme

Son tam yedekten sonra deęişen deęişiklikler yedeklenir. komutu řu řekildedir.

```
BACKUP DATABASE BLM4522  
TO DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\BLM4522_diff.bak' WITH DIFFERENTIAL;
```

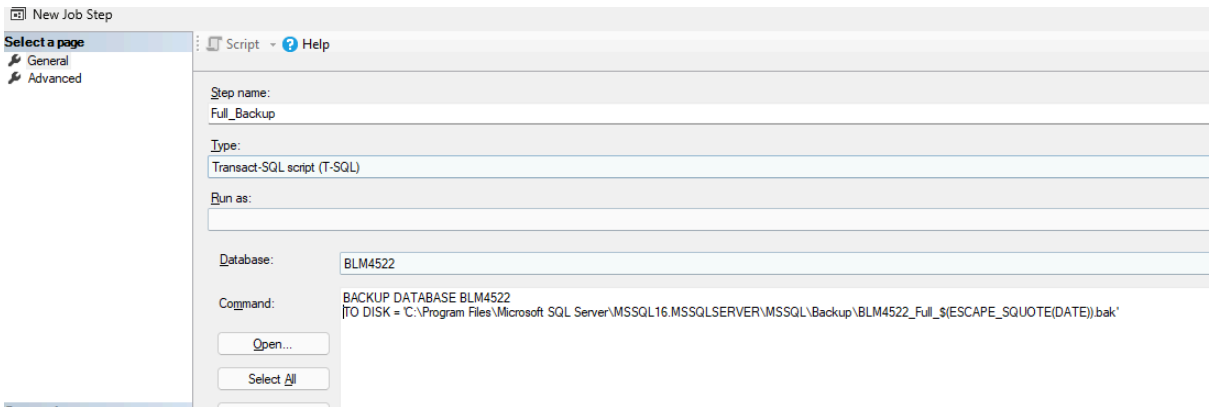
1.3. Artımlı Yedekleme

Son yedekten sonra deęişen deęişiklikler yedeklenir. Fark yedeklemeden farkı, fark yedekleme en son yedeklenen **tam** yedekten sonraki deęişikleri yedekler, artımlı yedekleme ise son yedeklemeden sonraki deęişiklikleri yedekler. Komutu řu řekildedir.

```
BACKUP LOG BLM4522  
TO DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\BLM4522_log.trn'
```

1.4. Zamanlayıcılarla Yedekleme

Zamanlayıcılarla yedekleme yapabilmek için SQL Server Agent aracılığı ile Job'lar oluşturulur. Job bölümünün step kısmına komutumuz girilir.



New Job Step

Select a page

- General
- Advanced

Script Help

Step name: Full_Backup

Type: Transact-SQL script (T-SQL)

Run as:

Database: BLM4522

Command: BACKUP DATABASE BLM4522
TO DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\BLM4522_Full_\$(ESCAPE_QUOTE(DATE)).bak'

Open...

Select All

Schedule kısmında hangi aralıklarla yedekleme yapılacağı belirlenir.

1.5. Felaketten Kurtarma Senaryoları

Bir veritabanındaki önemli bir tablo yanlışlıkla silindiğinde bu felaketi geri almak için tam yedekleme ve transaction log yedeği kullanılır.

a. Veritabanı Yedeğini Geri Yükleme (Full Backup)

Veritabanını tam yedeğinden geri yüklemek için aşağıdaki komut kullanılır.

```
RESTORE DATABASE BLM4522  
FROM DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\BLM4522_full.bak'  
WITH REPLACE;
```

Eğer veritabanındaki bir veri silindiyseniz ve tam yedeği geri yüklediyseniz, ancak bu işlemin ardından yeni veriler de eklenmişse, Artımlı yedeği kullanılarak sadece silinen verileri geri getirebilirsiniz.

```
RESTORE LOG VeritabaniAdi  
FROM DISK = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\BLM4522_log.trn'  
WITH NORECOVERY;
```

1.6. Kaydedilen Yedeklerin Doğruluğunu Test Etme

Aşağıdaki komut sayesinde yedek dosyamızın bütünlüğünü kontrol edebiliriz.

```
RESTORE VERIFYONLY  
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\BLM4522_full.bak'
```

2. Veritabanı Yedekleme ve Otomasyon Çalışması

2.1. Veritabanı Yedekleme İşlemini Otomatikleştirmek

SQL Server Agent kullanarak yedekleme işlemini otomatikleştirme 1.4.'de detaylı bir şekilde anlatılmıştır.

2.2. T-SQL Kullanarak Yedekleme Raporları Oluşturma

Aşağıdaki sorgu sayesinde geçmiş yedekleme kayıtlarını listelenir. Bu sorgu ile hangi veritabanının, ne zaman yedeklendiği ve nereye kaydedildiği aşağıdaki şekildeki gibi detaylı olarak listelenir.

```

SELECT
    database_name,
    backup_start_date,
    backup_finish_date,
    physical_device_name,
    type AS backup_type
FROM msdb.dbo.backupset bs
JOIN msdb.dbo.backupmediafamily bmf
    ON bs.media_set_id = bmf.media_set_id
WHERE database_name = 'BLM4522'
ORDER BY backup_finish_date DESC;

```

100 %

Results

Messages

	database_name	backup_start_date	backup_finish_date	physical_device_name	backup_type
1	BLM4522	2025-04-25 00:36:36.000	2025-04-25 00:36:36.000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...	I
2	BLM4522	2025-04-25 00:33:27.000	2025-04-25 00:33:27.000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...	D
3	BLM4522	2025-04-25 00:33:08.000	2025-04-25 00:33:08.000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...	D
4	BLM4522	2025-04-25 00:28:58.000	2025-04-25 00:28:58.000	C:\Program Files\Microsoft SQL Server\MSSQL16.MSS...	D

2.3. Otomatik Yedekleme Uyarıları

Öncelikle Database Mail ayarlarımızı yaparız

Manage Existing Account

Choose the account to view, change, or delete.



Account name:

mail



Delete

Description:

Outgoing mail server (SMTP)

E-mail address:

skyout800@gmail.com

Display name:

berk

Reply e-mail:

skyout800@gmail.com

Server name:

smtp.gmail.com

Port number:

587

☒ This server requires a secure connection (SSL)

SMTP Authentication

☐ Windows Authentication using Database Engine service credentials

☒ Basic authentication

User name:

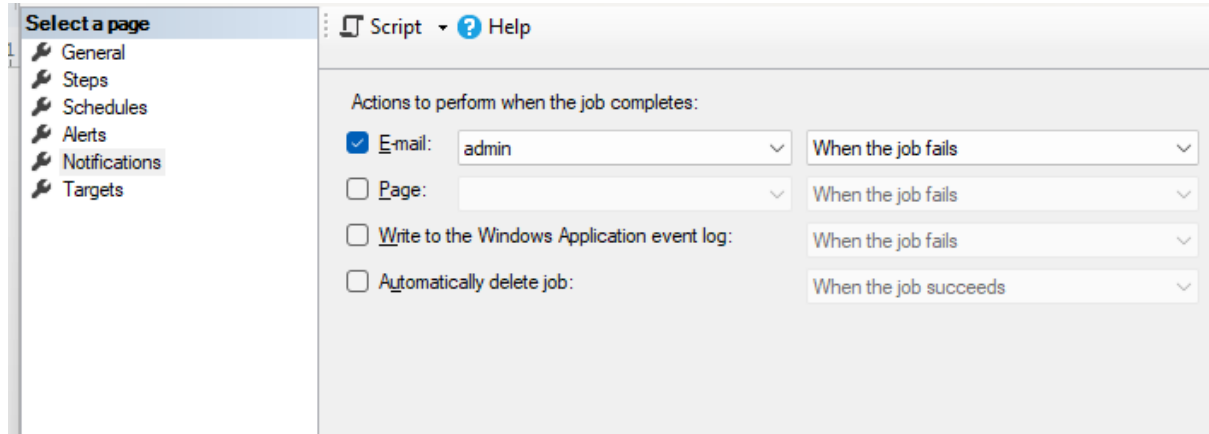
skyout800@gmail.com

Password:

Confirm password:

☐ Anonymous authentication

Ardından da Server Agent'a bunu tanıtmamız gerekir ki Mail sistemini kullanabilsin. Bunun için önce operator tanımlanır ve Job'umuzun properties kısmından "When the job fails" seçilir. Bu sayede işlemimizde hata oluşursa mail yoluyla bildirim gönderilir.



3. Veri tabanı Güvenliği ve Erişim Kontrolü

Bu bölüm veritabanı güvenliği ve erişim kontrolü konularını ele almaktadır. Genel olarak kullanıcı kimlik doğrulama yöntemleri, veri şifreleme, injection saldırılarına karşı koruma ve SQL Server Audit özelliklerinin kullanımından bahsedilecektir.

3.1. Erişim Kontrolü

Veritabanına erişimi SQL Server Authentication ve Windows Authentication ile sağlayabiliriz. SQL Server Authentication için öncelikle sunucuda oturum açabilecek kullanıcı oluşturulur, ardından da veritabanında bu kullanıcı oluşturulur.

```
CREATE LOGIN deneme_kullanici WITH PASSWORD = '1234';  
  
USE BLM4522;  
CREATE USER deneme_kullanici FOR LOGIN deneme_kullanici;
```

Bu kullanıcının erişebileceği tabloları belirleyebiliriz. Sadece istediğimiz tablolara erişim hakkı verip istediğimiz tablolara erişim hakkını yasaklayabiliriz.

```
GRANT SELECT on dbo.spend_analysis_dataset$ TO deneme_kullanici;  
DENY SELECT on dbo.spend_analysis_dataset$ TO deneme_kullanici;
```

GRANT SELECT, kullanıcıya o tabloya erişim hakkı tanır iken, DENY SELECT ise erişim hakkına izin vermez.

Windows Authentication kullanarak erişim sağlamak istiyor isek yukarı olduğu gibi bir kullanıcı ekleyip ardından oluşturmamız gerekmektedir.

```
CREATE LOGIN [berk\berk] FROM WINDOWS;  
USE BLM4522;  
CREATE USER [berk\berk] FOR LOGIN [berk\berk];
```

3.2. Veri Şifreleme

Veritabanının yetkisiz erişimlere karşı korunması amacıyla Transparent Data Encryption (TDE) yöntemi uygulanabilir. TDE, veritabanındaki tüm verilerin disk düzeyinde şifrlenmesini sağlar.

```
-- MASTER KEY oluşturulur (master veritabanında)  
USE master;  
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'SertifikaSifresi123!';  
  
-- Sertifika oluşturulur  
CREATE CERTIFICATE BLM4522_Cert  
WITH SUBJECT = 'BLM4522 Sertifikasi';  
  
-- Hedef veritabanına geçilir  
USE BLM4522;  
  
-- TDE için şifreleme anahtarı oluşturulur  
CREATE DATABASE ENCRYPTION KEY  
WITH ALGORITHM = AES_256  
ENCRYPTION BY SERVER CERTIFICATE BLM4522_Cert;  
  
-- Şifreleme etkinleştirilir  
ALTER DATABASE BLM4522 SET ENCRYPTION ON;
```


Öncelikle bir MASTER KEY oluşturulur, ardından sertifika oluşturulur. Son olarak da TDE için şifreleme anahtarı oluşturulur ve şifreleme etkinleştirilir.

3.3. SQL Injection Testleri

SQL Injection, kötü niyetli kullanıcıların web uygulamalarındaki veri giriş alanları aracılığıyla SQL sorgularına müdahale ederek veritabanına yetkisiz erişim sağlamaya çalıştığı bir saldırı türüdür.

```
SELECT * FROM dbo.user WHERE dbo.user = 'admin' AND sifre = '1234' OR 1=1;
```

Eğer bir kullanıcı yukarıdaki gibi bir sorgu girerse tüm user'lara erişebilir. Çünkü komuttaki OR 1=1 ifadesi her zaman doğru döndüreceğinden kimlik doğrulaması atlnaıp sistemdeki tüm kullanıcılar listelenebilir. SQL Injection'ı engellemenin en etkili yollarından biri parametrelili sorgular kullanmaktır. Bu yöntemle kullanıcıdan alınan veriler doğrudan sorguya gömülmez; veri ve sorgu mantıksal olarak ayrılır.

```
using (SqlCommand cmd = new SqlCommand("SELECT * FROM dbo.user WHERE username = @username AND password = @password", connection)){  
    cmd.Parameters.AddWithValue("@username", username);  
    cmd.Parameters.AddWithValue("@password", password);}
```

3.4. Audit Logları

Audit log, SQL Server'da yapılan işlemleri izleyip kayıt altına alan bir güvenlik mekanizmasıdır. Özellikle kim, ne zaman, hangi veritabanı nesnesi üzerinde hangi işlemi yaptı gibi bilgileri saklar. Bu sayede veritabanı yöneticisi, sistemde gerçekleşen olayları sonradan inceleyebilir. Örneğin bir tablo silindiyse, bunun hangi kullanıcı tarafından ve hangi tarihte yapıldığını öğrenmek mümkündür.

```
--Audit Özelliğini Açarız  
CREATE SERVER AUDIT BLM4522_Audit  
TO FILE (FILEPATH = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\AuditLogs\');
```

Yukarıda Audit loglarının yazılacağı dosya konumunu belirten bir audit nesnesi oluşturulmuştur.

Ardından Audit'i etkinleştiririz.

```
-- Audit'i etkinleştiririz  
ALTER SERVER AUDIT BLM4522_Audit WITH (STATE = ON);
```

Aşağıdaki ifade sayesinde de tablomuzda yapılan bütün SELECT ve INSERT işlemlerinin hareketlerini izleme özelliği tanımlarız.

```
-- Belirli hareketleri izlemek için AUDIT SPECIFICATION tanımlarız  
CREATE DATABASE AUDIT SPECIFICATION BLM4522_DBSpec  
FOR SERVER AUDIT BLM4522_Audit  
ADD (SELECT ON OBJECT::spend_analysis_dataset$ BY [public]),  
ADD (INSERT ON spend_analysis_dataset$ BY [public]);
```

Son olarak da Audit Spec'imizi aktif hale getiririz.

```
ALTER DATABASE AUDIT SPECIFICATION BLM4522_DBSpec WITH (STATE = ON);
```

Altteki komut ile de Audit kayıtlarını okuyup inceleyebiliriz.

```
SELECT * FROM sys.fn_get_audit_file  
('C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\AuditLogs\*', NULL, NULL);
```