

Mustafa Girgin 21290649

<https://github.com/antelcha/sqlprojeler>

Otomatik Yedekleme Sistemi Kurulumu ve Testi

Bu projede, SQL Server veritabanları için kapsamlı bir otomatik yedekleme sistemi geliştirdim. Sistem, düzenli yedeklemeleri otomatik olarak alıp, yedekleme geçmişini takip ediyor ve olası hataları raporluyor.

1. Veritabanı ve Tablo Yapılarının Oluşturulması

İlk adım olarak, yedekleme sisteminin ihtiyaç duyduğu veritabanı ve tabloları oluşturdum.

TestDB adında bir test veritabanı ve içerisinde örnek veriler için Customers tablosu ile yedekleme kayıtları için BackupHistory tablosu yer alıyor.

```
-- Veritabanı kontrolü ve oluşturma
IF NOT EXISTS (SELECT name FROM sys.databases WHERE name = 'TestDB')
BEGIN
    CREATE DATABASE TestDB;
    PRINT 'TestDB veritabanı oluşturuldu.';
END
ELSE
BEGIN
    PRINT 'TestDB veritabanı zaten mevcut.';
END
GO
```

```
USE TestDB;
GO

-- Customers tablosu kontrolü ve oluşturma
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[Customers]') AND type in (N'U'))
BEGIN
    CREATE TABLE Customers (
        CustomerID INT PRIMARY KEY IDENTITY(1,1),
        FirstName NVARCHAR(50),
        LastName NVARCHAR(50),
        Email NVARCHAR(100),
        CreatedDate DATETIME DEFAULT GETDATE()
    );
    PRINT 'Customers tablosu oluşturuldu.';
END
```

```

ELSE
BEGIN
    PRINT 'Customers tablosu zaten mevcut.';
END
GO

-- BackupHistory tablosu kontrolü ve oluşturma
IF NOT EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[BackupHistory]') AND type in (N'U'))
BEGIN
    CREATE TABLE BackupHistory (
        BackupID INT PRIMARY KEY IDENTITY(1,1),
        DatabaseName NVARCHAR(100),
        BackupType NVARCHAR(50),
        BackupStartTime DATETIME,
        BackupEndTime DATETIME,
        BackupStatus NVARCHAR(50),
        BackupSizeKB BIGINT,
        BackupLocation NVARCHAR(500)
    );
    PRINT 'BackupHistory tablosu oluşturuldu.';
END
ELSE
BEGIN
    PRINT 'BackupHistory tablosu zaten mevcut.';
END
GO

-- Örnek veri ekleme
IF NOT EXISTS (SELECT TOP 1 * FROM Customers)
BEGIN
    INSERT INTO Customers (FirstName, LastName, Email)
    VALUES
        ('John', 'Doe', 'john@example.com'),
        ('Jane', 'Smith', 'jane@example.com'),
        ('Mike', 'Johnson', 'mike@example.com');
    PRINT 'Örnek müşteri verileri eklendi.';
END
ELSE
BEGIN
    PRINT 'Customers tablosunda zaten veri mevcut.';
END
GO

```

2. Yedekleme Prosedürünün Oluşturulması

Yedekleme işlemlerini yönetmek için sp_CreateBackup adında bir stored procedure oluşturdum.

Bu prosedür:

Yedekleme başlangıç kaydını oluşturuyor
Yedekleme işlemini gerçekleştiriyor
İşlem sonucunu kaydediyor
Hata durumunda uygun şekilde raporluyor

```
USE TestDB;
GO

-- Yedekleme prosedürü kontrolü ve oluşturma
IF EXISTS (SELECT * FROM sys.objects WHERE object_id =
OBJECT_ID(N'[dbo].[sp_CreateBackup]') AND type in (N'P'))
BEGIN
    DROP PROCEDURE [dbo].[sp_CreateBackup];
    PRINT 'Eski sp_CreateBackup prosedürü silindi.';
END
GO

CREATE PROCEDURE sp_CreateBackup
    @DatabaseName NVARCHAR(100),
    @BackupPath NVARCHAR(500)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @BackupFileName NVARCHAR(1000)
    DECLARE @StartTime DATETIME
    DECLARE @SQL NVARCHAR(MAX)

    SET @StartTime = GETDATE()
    SET @BackupFileName = @BackupPath + @DatabaseName + '_' +
        REPLACE(CONVERT(VARCHAR, @StartTime, 112) + REPLACE(CONVERT(VARCHAR,
@StartTime, 108), ':', ''), ' ', '_') + '.bak'

    BEGIN TRY
        -- Yedekleme başlangıç kaydı
        INSERT INTO BackupHistory (DatabaseName, BackupType, BackupStartTime,
BackupStatus)
VALUES (@DatabaseName, 'FULL', @StartTime, 'IN_PROGRESS')

        -- Yedekleme komutunu oluştur
        SET @SQL = 'BACKUP DATABASE ' + @DatabaseName +
            ' TO DISK = ''' + @BackupFileName + ''''

        -- Yedekleme işlemini gerçekleştir
        EXEC (@SQL)

        -- Yedekleme kaydını güncelle
```

```

        UPDATE BackupHistory
        SET BackupEndTime = GETDATE(),
            BackupStatus = 'COMPLETED',
            BackupLocation = @BackupFileName,
            BackupSizeKB = (SELECT size * 8 FROM sys.master_files WHERE
database_id = DB_ID(@DatabaseName) AND type_desc = 'ROWS')
        WHERE DatabaseName = @DatabaseName
        AND BackupStartTime = @StartTime

        PRINT 'Yedekleme işlemi başarıyla tamamlandı.';
    END TRY
    BEGIN CATCH
        -- Hata durumunda kaydı güncelle
        UPDATE BackupHistory
        SET BackupEndTime = GETDATE(),
            BackupStatus = 'FAILED'
        WHERE DatabaseName = @DatabaseName
        AND BackupStartTime = @StartTime

        PRINT 'Hata: ' + ERROR_MESSAGE();
        THROW;
    END CATCH
END;
GO
PRINT 'sp_CreateBackup prosedürü oluşturuldu.';

-- Yedekleme rapor view kontrolü ve oluşturma
IF EXISTS (SELECT * FROM sys.views WHERE object_id =
OBJECT_ID(N'[dbo].[vw_BackupReport]'))
BEGIN
    DROP VIEW [dbo].[vw_BackupReport];
    PRINT 'Eski vw_BackupReport view silindi.';
END
GO

CREATE VIEW vw_BackupReport
AS
SELECT
    DatabaseName,
    BackupType,
    BackupStartTime,
    BackupEndTime,
    DATEDIFF(MINUTE, BackupStartTime, BackupEndTime) as DurationMinutes,
    BackupStatus,
    BackupSizeKB / 1024.0 as BackupSizeMB,
    BackupLocation
FROM BackupHistory;
GO
PRINT 'vw_BackupReport view oluşturuldu.';

```

3. Otomatik Yedekleme Job'ının Oluşturulması

SQL Server Agent kullanarak Daily_Database_Backup adında bir job oluşturdum. Bu job her gece yarısı otomatik olarak çalışacak şekilde ayarlandı.

```
USE msdb;
GO

-- Job kontrolü ve silme
IF EXISTS (SELECT job_id FROM msdb.dbo.sysjobs WHERE name =
N'Daily_Database_Backup')
BEGIN
    EXEC dbo.sp_delete_job @job_name = N'Daily_Database_Backup';
    PRINT 'Eski Daily_Database_Backup job silindi.';
END
GO

-- Schedule kontrolü ve silme
IF EXISTS (SELECT schedule_id FROM msdb.dbo.sysschedules WHERE name =
N'DailyBackupSchedule')
BEGIN
    EXEC dbo.sp_delete_schedule @schedule_name = N'DailyBackupSchedule';
    PRINT 'Eski DailyBackupSchedule schedule silindi.';
END
GO

BEGIN TRY
    -- Job oluştur
    EXEC dbo.sp_add_job
        @job_name = N'Daily_Database_Backup',
        @description = N'Her gün otomatik yedek alma işlemi',
        @enabled = 1;

    -- Job'a adım ekle
    EXEC sp_add_jobstep
        @job_name = N'Daily_Database_Backup',
        @step_name = N'Perform Backup',
        @subsystem = N'TSQL',
        @command = N'EXEC TestDB.dbo.sp_CreateBackup
                    @DatabaseName = ''TestDB'',
                    @BackupPath = ''/var/opt/mssql/backup/'';

    -- Günlük çalışma planı oluştur
    EXEC dbo.sp_add_schedule
        @schedule_name = N'DailyBackupSchedule',
        @freq_type = 4, -- Günlük
        @freq_interval = 1, -- Her gün
```

```

        @active_start_time = 000000; -- Gece yarısı (00:00)

-- Schedule'ı job'a bağla
EXEC sp_attach_schedule
    @job_name = N'Daily_Database_Backup',
    @schedule_name = N'DailyBackupSchedule';

-- Job'ı aktif et
EXEC dbo.sp_add_jobserver
    @job_name = N'Daily_Database_Backup';

PRINT 'Daily_Database_Backup job başarıyla oluşturuldu.';
END TRY
BEGIN CATCH
    PRINT 'Hata: ' + ERROR_MESSAGE();
    THROW;
END CATCH
GO

```

```

[2025-05-29 21:01:56] [S0001][22022] SQLServerAgent
Daily_Database_Backup job basariyla olusturuldu.
[2025-05-29 21:01:56] completed in 41 ms

```

4. Test ve Doğrulama

Sistemin düzgün çalıştığından emin olmak için çeşitli testler gerçekleştirdim:

Manuel Yedekleme Testi

İlk olarak manuel bir yedekleme işlemi başlattım:

```

USE TestDB;
GO

-- Manuel yedek alma testi
PRINT 'Manuel yedekleme testi başlatılıyor...';
EXEC sp_CreateBackup
    @DatabaseName = 'TestDB',
    @BackupPath = '/var/opt/mssql/backup/';
GO

-- Yeni test verisi ekleme
PRINT 'Test verileri ekleniyor...';
INSERT INTO Customers (FirstName, LastName, Email)

```

```

VALUES
    ('Sarah', 'Wilson', 'sarah@example.com'),
    ('Robert', 'Brown', 'robert@example.com');
GO

-- Job durumunu kontrol et
USE msdb;
GO

PRINT 'Job durumu kontrol ediliyor...';
SELECT
    j.name AS 'Job Name',
    CASE j.enabled
        WHEN 1 THEN 'Enabled'
        ELSE 'Disabled'
    END AS 'Job Status',
    CASE
        WHEN jh.run_date IS NOT NULL AND jh.run_time IS NOT NULL
        THEN CONVERT(DATETIME,
            CAST(jh.run_date AS CHAR(8)) + ' ' +
            STUFF(STUFF(RIGHT('000000' + CAST(jh.run_time AS VARCHAR(6)), 6), 5,
0, ':'), 3, 0, ':'))
        END AS 'Last Run Time',
    CASE jh.run_status
        WHEN 0 THEN 'Failed'
        WHEN 1 THEN 'Succeeded'
        WHEN 2 THEN 'Retry'
        WHEN 3 THEN 'Canceled'
        WHEN 4 THEN 'In Progress'
    END AS 'Last Run Status'
FROM
    msdb.dbo.sysjobs j
    LEFT JOIN (
        SELECT
            job_id,
            run_date,
            run_time,
            run_status
        FROM msdb.dbo.sysjobhistory
        WHERE step_id = 0
    ) jh ON j.job_id = jh.job_id
WHERE
    j.name = 'Daily_Database_Backup';
GO

-- Yedekleme geçmişini kontrol et
USE TestDB;
GO

```

```

PRINT 'Yedekleme geçmişi kontrol ediliyor...';
SELECT
    DatabaseName,
    BackupType,
    FORMAT(BackupStartTime, 'yyyy-MM-dd HH:mm:ss') as StartTime,
    FORMAT(BackupEndTime, 'yyyy-MM-dd HH:mm:ss') as EndTime,
    DurationMinutes,
    BackupStatus,
    CAST(BackupSizeMB AS DECIMAL(10,2)) as BackupSizeMB,
    BackupLocation
FROM vw_BackupReport
ORDER BY BackupStartTime DESC;
GO

-- Job'ı manuel olarak çalıştır
USE msdb;
GO

PRINT 'Job manuel olarak çalıştırılıyor...';
EXEC msdb.dbo.sp_start_job N'Daily_Database_Backup';
GO

-- 10 saniye bekle
WAITFOR DELAY '00:00:10';
GO

-- Son durumu tekrar kontrol et
PRINT 'Son durum kontrol ediliyor...';
USE TestDB;
GO

SELECT TOP 5 * FROM vw_BackupReport
ORDER BY BackupStartTime DESC;
GO

```

Bu proje sayesinde, veritabanı yedeklemelerinin güvenli ve otomatik bir şekilde alınmasını, takip edilmesini ve raporlanmasını sağlayan kapsamlı bir sistem geliştirmiş oldum. Sistem, olası hata durumlarına karşı dayanıklı ve kolayca izlenebilir bir yapıda tasarlandı.