

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM4522 AĞ TABANLI PARALEL DAĞITIM SİSTEMLERİ FINAL ÖDEVİ**

**Emre Ceniklioğlu-20290234**

**Berkan Bucak-21290438**

**02.07.2025**

# LINKLER

**Github Repo Linki:** <https://github.com/emreceniklioglu/MS-SQL-final-project>

**Videolar İçin Drive Linki:**

<https://drive.google.com/drive/folders/125CszS7vmdrYczQ1d2BOo0DcGavd3jDT?usp=sharing>

**kullanılan veri tabanlarını indirme linkleri :**

<https://github.com/Microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorks2022.bak>

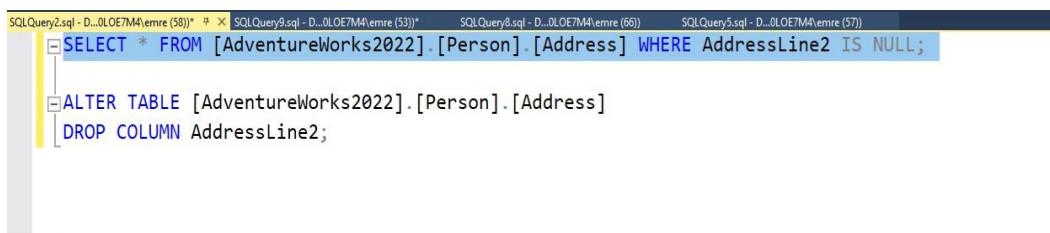
<https://github.com/Microsoft/sql-server-samples/releases/download/adventureworks/AdventureWorksLT2022.bak>

# 1. Giriş

Bu raporda, Microsoft SQL Server ortamında *AdventureWorks2022* veritabanı kullanılarak gerçekleştirilen çeşitli veri işleme ve yönetimi uygulamaları ele alınmıştır. ETL (Extract, Transform, Load) süreçlerinin tasarımı, veritabanı performansının izlenmesi ve iyileştirilmesi, veri yönetici rolleri ile veritabanı yedekleme ve otomasyon işlemleri gibi konular uygulamalı örneklerle detaylandırılmıştır. Gerçekleştirilen çalışmalar, teorik bilgilerin pratik uygulamalarla pekiştirilmesini sağlamış ve SQL tabanlı veri yönetim becerilerinin geliştirilmesine katkı sunmuştur.

## 2. Veri Temizleme ve ETL Süreçleri Tasarımı

### a. Veri Temizleme



```
SELECT * FROM [AdventureWorks2022].[Person].[Address] WHERE AddressLine2 IS NULL;

ALTER TABLE [AdventureWorks2022].[Person].[Address]
DROP COLUMN AddressLine2;
```

AddressID	AddressLine1	AddressLine2	City	StateProvinceID	PostalCode	SpatialLocation	rowguid	ModifiedDate
95	2639 Anchor Court	NULL	Seattle	79	98104	0xE6100000010CDAFB0D9327D5474067D6CE07828C5E...	9C3D00B0-E3B3-4696-B0EB-A90356CB9ACC	2014-02-07 00:00:00.000
96	502 Alexander Pl.	NULL	Seattle	79	98104	0xE6100000010C528340978BD4740703AEFB51E95EC0	B3D67582-A180-42AD-BE0D-E8FAC4080354	2009-01-25 00:00:00.000
97	5802 Ampersand Dr...	NULL	Seattle	79	98104	0xE6100000010C63447B90C3D54740703AEFB51E95EC0	137CC1BA-089A-4E2D-B1B7-7480B02A7E00	2008-12-31 00:00:00.000
98	100 5125 Cotton Ct.	NULL	Seattle	79	98104	0xE6100000010CEB8C18805FD647406492C68FAF965EC0	AB68CC20-67D3-4A26-BA02-CD917C2DE...	2009-02-03 00:00:00.000
99	103 9537 Ridgewood Dr...	NULL	Seattle	79	98104	0xE6100000010CA809F7D5ABC84740593EB2FC7975EC0	905F7B49-E609-42C8-B9F5-ED5C043D16C	2010-02-05 00:00:00.000
100	104 9564 North Ridge D...	NULL	Seattle	79	98104	0xE6100000010C0F656B2A0CF4740868A4002F0915EC0...	38560935-AD49-458D-9C5B-3D09C4B1E3C	2009-01-29 00:00:00.000
101	105 1619 Stillman Court	NULL	Seattle	79	98104	0xE6100000010CDAFB0D9327D547406A6DAA2E2795E...	3A7BAABE-A265-43DE-86E1-CFE6EA60E3B0	2008-12-19 00:00:00.000
102	106 2144 San Rafael	NULL	Seattle	79	98104	0xE6100000010C9787F3A4E7D0A740868A4002F0915EC0	E0D82A90-FAE8-43A9-A907-F6512E57709	2009-01-05 00:00:00.000
103	107 7403 N. Broadway	NULL	Seattle	79	98104	0xE6100000010CEB8C18805FD647407BE217CDB935EC0	C9F9E90D9-6857-44FC-C5A0-7D6C9C6A6FF	2009-01-01 00:00:00.000
104	108 7842 Ygnacio Valley...	NULL	Seattle	79	98104	0xE6100000010CA809F7D5ABC8474056D9A14ED88E5E...	6600064E-0821-405D-85CA-6A5E1E41771D	2008-12-10 00:00:00.000
105	109 874 Oliver Road	NULL	Seattle	79	98104	0xE6100000010CB90968A1B7D24740703AEFB51E955EC0	F868AAE1-DACB-4606-89C4-E807F4492629	2010-02-24 00:00:00.000
106	110 1064 Slow Creek R...	NULL	Seattle	79	98104	0xE6100000010CEB8C18805FD647407DF22377F295EC0	184AD4A1-45B8-4910-A2B2-A9E2A658B07C	2008-12-08 00:00:00.000
107	111 77 Birchwood	NULL	Seattle	79	98104	0xE6100000010C22C0590EBFC7474051867D0B8A38F5EC0	C27E3842-1548-404C-B104-F765CF42C068	2009-01-27 00:00:00.000
108	112 7765 Sunaine Drive	NULL	Seattle	79	98104	0xE6100000010CEB8C18805FD647404D6E54954915EC0	40E9167A-D85B-4D4B-9B57-F5030A37E4AC	2009-02-19 00:00:00.000
109	113 1102 Ravenwood	NULL	Seattle	79	98104	0xE6100000010CDAFB0D9327D547409A689D608995EC0	BA12573D-D7C2-4C04-BB1A-2F050626670	2009-12-15 00:00:00.000
110	114 1398 Yorba Linda	NULL	Seattle	79	98104	0xE6100000010C4122069E53D347408EE60B0999A5EC0	C32CA551-349E-42FD-A80B-6170E770B516	2009-02-08 00:00:00.000
111	115 4948 W 4th St	NULL	Seattle	79	98104	0xF0F100000010C000911ARFVN0N1A914FNRF8FF0N	F377R01211FFD7442-A147-477A-8F5877701	2009-03-11 00:00:00.000

Sekil 1

Sekil 1'de görüldüğü üzere veritabanımızdaki bu tabloda AddressLine2 isimli sütündaki verilerin %90'ı NULL olarak atanmıştır, bu sütun silinerek tablo sadeleştirilmiştir.

```

SELECT * FROM [AdventureWorks2022].[Person].[Address] WHERE AddressLine2 IS NULL;

-- 1. Önce bağımlı indeksi sil
DROP INDEX IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode
ON [AdventureWorks2022].[Person].[Address];

-- 2. Sütunu sili
ALTER TABLE [AdventureWorks2022].[Person].[Address]
DROP COLUMN AddressLine2;

SELECT top(100)* FROM [AdventureWorks2022].[Person].[Address];

```

Results Spatial results Messages

AddressID	AddressLine1	AddressLine2	City	StateProvinceID	PostalCode	SpatialLocation	rowguid	ModifiedDate
1	1970 Napa Ct.	NULL	Bothell	79	98011	0xE6100000010CAE8BFC28BCE4474067A89189898A5EC0	9AACDB0D-36CF-483F-84D8-585C2D4EC6E9	2007-12-04 00:00:00.000
2	9833 Mt. Dias Blv.	NULL	Bothell	79	98011	0xE6100000010CD6FA851AE6D74740BC262A0A03905EC0	32A54B9E-E034-4FB8-B573-A71CDE60D8C0	2008-11-30 00:00:00.000
3	7484 Roundtree Drive	NULL	Bothell	79	98011	0xE6100000010C18E304C4ADE14740DA930C7893915EC0	4C506923-D1B4-452c-A07C-BAAF5B142A4	2013-03-07 00:00:00.000
4	9539 Glenside Dr	NULL	Bothell	79	98011	0xE6100000010C813AD05FFDE474011A5C28A7C955EC0	E5946C78-4BCC-477F-9FA1-CC09DE16A880	2009-02-03 00:00:00.000
5	1226 Shoe St.	NULL	Bothell	79	98011	0xE6100000010C61C64D8ABBD94740C460EA3FD8855EC0	FBAFF937-4A97-4AF0-81FD-B849900E9BB0	2008-12-19 00:00:00.000
6	1399 Firestone Drive	NULL	Bothell	79	98011	0xE6100000010CCE0B4E5048DAA7402F12A5F0C975EC0	FEFB8191-9804-4AC8-877A-33FDE34F0075	2009-02-13 00:00:00.000
7	5672 Hale Dr.	NULL	Bothell	79	98011	0xE6100000010C18E304C4ADE1474011A5C28A7C955EC0	0175A174-C34-4D41-B3C1-4419CD6A0446	2009-12-11 00:00:00.000
8	6387 Scenic Avenue	NULL	Bothell	79	98011	0xE6100000010C0D29A5C9381E7A70C62A0C975EC0	2715E813-4DCA-49E1-8E1C-31857D21E593	2008-12-17 00:00:00.000

Şekil 2

Şekil 2'de NULL olan AddressLine2 sütunu silinirken yazılmış kodlar gösterilmiştir.

```

-- 1. Önce bağımlı indeksi sil
DROP INDEX IX_Address_AddressLine1_AddressLine2_City_StateProvinceID_PostalCode
ON [AdventureWorks2022].[Person].[Address];

-- 2. Sütunu sili
ALTER TABLE [AdventureWorks2022].[Person].[Address]
DROP COLUMN AddressLine2;

SELECT top(100)* FROM [AdventureWorks2022].[Person].[Address];

```

Results Spatial results Messages

AddressID	AddressLine1	City	StateProvinceID	PostalCode	SpatialLocation	rowguid	ModifiedDate
1	1970 Napa Ct.	Bothell	79	98011	0xE6100000010CAE8BFC28BCE4474067A89189898A5EC0	9AACDB0D-36CF-483F-84D8-585C2D4EC6E9	2007-12-04 00:00:00.000
2	9833 Mt. Dias Blv.	Bothell	79	98011	0xE6100000010CD6FA851AE6D74740BC262A0A03905EC0	32A54B9E-E034-4FB8-B573-A71CDE60D8C0	2008-11-30 00:00:00.000
3	7484 Roundtree Drive	Bothell	79	98011	0xE6100000010C18E304C4ADE14740DA930C7893915EC0	4C506923-D1B4-452c-A07C-BAAF5B142A4	2013-03-07 00:00:00.000
4	9539 Glenside Dr	Bothell	79	98011	0xE6100000010C813AD05FFDE474011A5C28A7C955EC0	E5946C78-4BCC-477F-9FA1-CC09DE16A880	2009-02-03 00:00:00.000
5	1226 Shoe St.	Bothell	79	98011	0xE6100000010C61C64D8ABBD94740C460EA3FD8855EC0	FBAFF937-4A97-4AF0-81FD-B849900E9BB0	2008-12-19 00:00:00.000
6	1399 Firestone Drive	Bothell	79	98011	0xE6100000010CCE0B4E5048DAA7402F12A5F0C975EC0	FEFB8191-9804-4AC8-877A-33FDE34F0075	2009-02-13 00:00:00.000
7	5672 Hale Dr.	Bothell	79	98011	0xE6100000010C18E304C4ADE1474011A5C28A7C955EC0	0175A174-C34-4D41-B3C1-4419CD6A0446	2009-12-11 00:00:00.000
8	6387 Scenic Avenue	Bothell	79	98011	0xE6100000010C0D29A5C9381E7A70C62A0C975EC0	2715E813-4DCA-49E1-8E1C-31857D21E593	2008-12-17 00:00:00.000
9	8713 Yosemite Ct.	Bothell	79	98011	0xE6100000010C6GA80AD742DDC474040862564B7825EC0	268AF621-76D7-4C78-9441-144FD139821A	2012-05-30 00:00:00.000
10	250 Race Court	Bothell	79	98011	0xE6100000010C19D64AE1FE4474040862564B7825EC0	06B6739D-8EB6-4378-8D55-FE196AF34C04	2008-12-02 00:00:00.000
11	1318 Lasalle Street	Bothell	79	98011	0xE6100000010CCEFBFD9FE947057B9479C7285EC0	981B3303-ACA2-49C7-9A96-F670785B269	2013-02-28 00:00:00.000
12	5415 San Gabriel Dr.	Bothell	79	98011	0xE6100000010C6GA80AD742DDC474011A5C28A7C955EC0	1C2C9CFE-AB9F-4F96-8E1F-D9666B6F7F22	2013-01-05 00:00:00.000

Şekil 3

Şekil 3'te AdressLine2 sütunu silindiğinden sonra tablonun durumu gösterilmiştir.

## b. Veri Dönüştürme

```
SELECT * FROM [AdventureWorks2022].[Person].[PersonPhone] WHERE LEN(PhoneNumber) != 12;

UPDATE [AdventureWorks2022].[Person].[PersonPhone]
SET PhoneNumber = LTRIM(SUBSTRING(PhoneNumber, CHARINDEX(')', PhoneNumber) + 1, LEN(PhoneNumber)))
WHERE PhoneNumber LIKE '1 (%)' AND LEN(PhoneNumber) != 12;

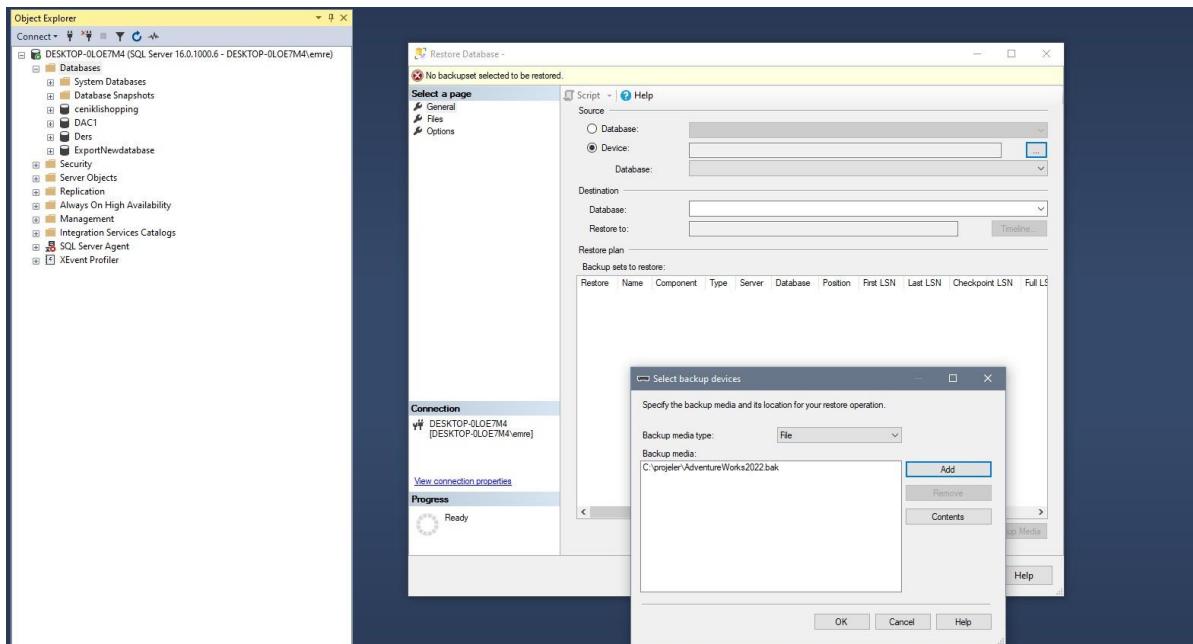
SELECT * FROM [AdventureWorks2022].[Person].[PersonPhone] WHERE LEN(PhoneNumber) != 12;
```

BusinessEntityID	PhoneNumber	PhoneNumberTypeID	ModifiedDate
1	1 (11) 500 555-0190	1	2013-05-23 00:00:00.000
2	1 (11) 500 555-0140	1	2013-05-23 00:00:00.000
3	1 (11) 500 555-0145	3	2012-05-23 00:00:00.000
4	1 (11) 500 555-0117	1	2012-05-23 00:00:00.000
5	1 (11) 500 555-0132	1	2015-04-15 16:31:33.090
6	1 (11) 500 555-0198	1	2013-06-30 00:00:00.000
7	1 (11) 500 555-0150	1	2012-07-31 00:00:00.000
8	1 (11) 500 555-0190	1	2013-07-31 00:00:00.000

Şekil 4

Şekil 4'te görüldüğü üzere önce alan kodu olan telefon numaraları (yani 12 karakterden fazla olanlar) seçilmiş, sonra bu numaraların alan kodu kaldırılarak bütün telefon numaraları aynı formata dönüştürülmüştür. 12 karakterden uzun olan numaraları bir daha çağrıdığımızda artık kalmadığı görülmektedir.

## c. Veri Yükleme



Şekil 5

Şekil 5'te AdventureWorks veritabanı mssql uygulamasına yüklenmiştir.

The screenshot shows a SQL Server Management Studio (SSMS) interface. In the top pane, there is a query window containing the following T-SQL code:

```
SELECT * FROM [AdventureWorks2022].[Purchasing].[ShipMethod]
INSERT INTO [Purchasing].[ShipMethod] ([Name],[ShipBase],[ShipRate],[ModifiedDate])
VALUES
    ('CARGO-X',10.55,1.34,'2025-04-22 00:00:00.000')
SELECT * FROM [AdventureWorks2022].[Purchasing].[ShipMethod]
```

The bottom pane displays two result grids. The first grid shows the initial state of the ShipMethod table:

ShipMethodID	Name	ShipBase	ShipRate	rowguid	ModifiedDate
1	XRQ - TRUCK GROUND	3.95	0.99	6BE756D9-D7BE-4463-8F2C-AE60C710D606	2008-04-30 00:00:00.000
2	ZY - EXPRESS	9.95	1.99	3455079B-F773-4DC6-8F1E-2A58649C4AB8	2008-04-30 00:00:00.000
3	OVERSEAS - DELUXE	29.95	2.99	22F4E461-28CF-4ACE-A980-F686CF112EC8	2008-04-30 00:00:00.000
4	OVERNIGHT J-FAST	21.95	1.29	107E8356-E7A8-463D-B60C-079FFF467F3F	2008-04-30 00:00:00.000
5	CARGO TRANSPORT 5	8.99	1.49	B166019A-B134-4E76-B957-B0A90C610ED	2008-04-30 00:00:00.000

The second grid shows the state of the table after the new row was inserted:

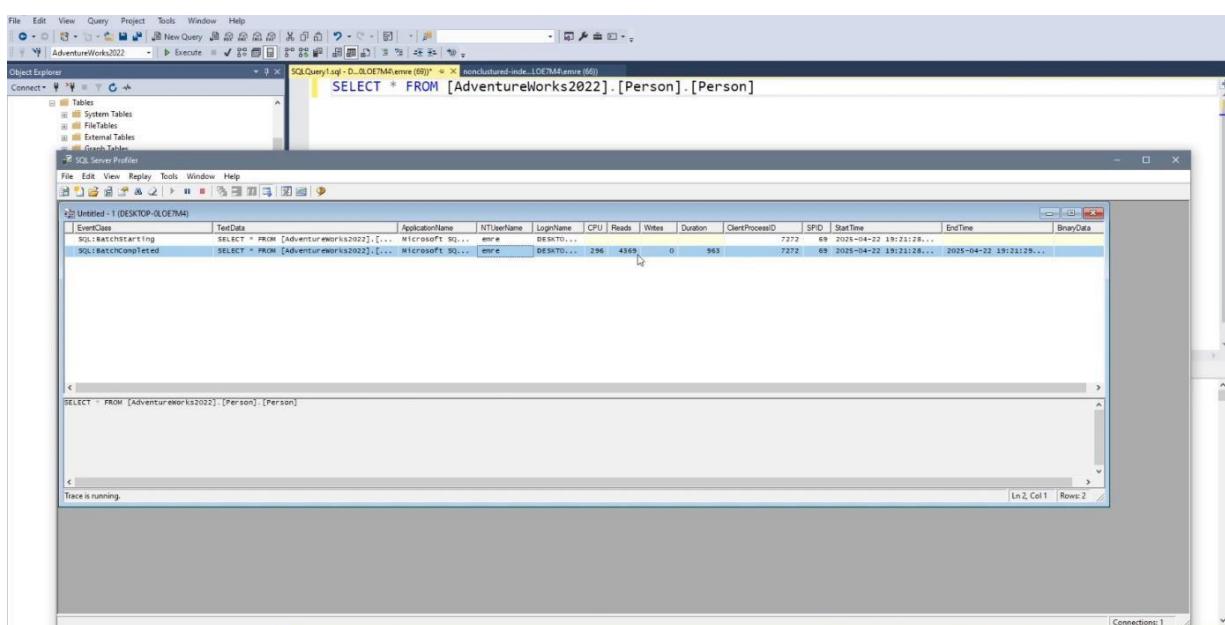
ShipMethodID	Name	ShipBase	ShipRate	rowguid	ModifiedDate
1	XRQ - TRUCK GROUND	3.95	0.99	6BE756D9-D7BE-4463-8F2C-AE60C710D606	2008-04-30 00:00:00.000
2	ZY - EXPRESS	9.95	1.99	3455079B-F773-4DC6-8F1E-2A58649C4AB8	2008-04-30 00:00:00.000
3	OVERSEAS - DELUXE	29.95	2.99	22F4E461-28CF-4ACE-A980-F686CF112EC8	2008-04-30 00:00:00.000
4	OVERNIGHT J-FAST	21.95	1.29	107E8356-E7A8-463D-B60C-079FFF467F3F	2008-04-30 00:00:00.000
5	CARGO TRANSPORT 5	8.99	1.49	B166019A-B134-4E76-B957-B0A90C610ED	2008-04-30 00:00:00.000
6	CARGO-X	10.55	1.34	2C293BF2-40F9-4C80-B7A1-ED4C8170AFF	2025-04-22 00:00:00.000

Şekil 6

Şekil 6'da veritabanına insert koduyla tabloya yeni veri eklenmiştir.

### 3. Veritabanı Performans Optimizasyonu ve İzleme

#### a. Veritabanı İzleme



Şekil 7

Şekil 7'de bir veritabanında bir sorgu çalıştırıldıkten sonra SQL Server Profiler açılıp sorgu hakkında bilgilere bakılmaktadır. Şekilde görüldüğü gibi SQL Server Profiler üzerinden sorgu hakkında cpu kullanımı, okuma sayısı, süresi, başlangıç ve bitiş zamanı gibi bilgilere ulaşılabilmektedir.

## b. Index Yönetimi

The screenshot shows two windows side-by-side. On the left is the Object Explorer of SQL Server Management Studio, which lists various database objects like Production, Sales, and HumanResources. In the center is a query window containing two T-SQL scripts:

```
-- ProductID Üzerinde nonclustered index olmadan sorgu
SELECT * FROM [Adventureworks2022].[Production].[TransactionHistory]
WHERE ProductID = 879;

-- ProductID sütunu Üzerinde nonclustered index oluşturup yapılan sorgu
CREATE NONCLUSTERED INDEX index_Productid ON [Production].[TransactionHistory] (ProductID);
```

On the right is the SQL Server Profiler window titled "Untitled - 1 DESKTOP-OLDETMA". It displays a trace of events and their details. The trace includes rows for connection events, batch starting, and batch completed. The last few rows show the execution of the first query, with the duration being approximately 0 seconds and CPU usage being 0.

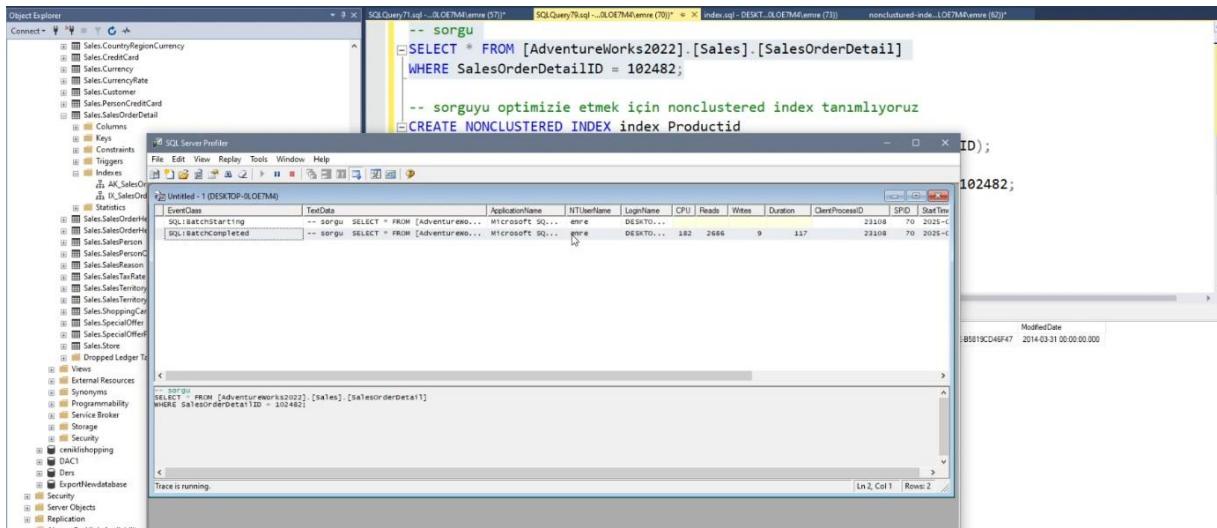
Şekil 8

This screenshot is similar to the one above, but it shows the results of the query execution after the index has been created. The SQL Server Profiler window now shows a significantly longer duration for the execution of the query, indicating the overhead of reading from the nonclustered index. The duration is listed as 55 seconds.

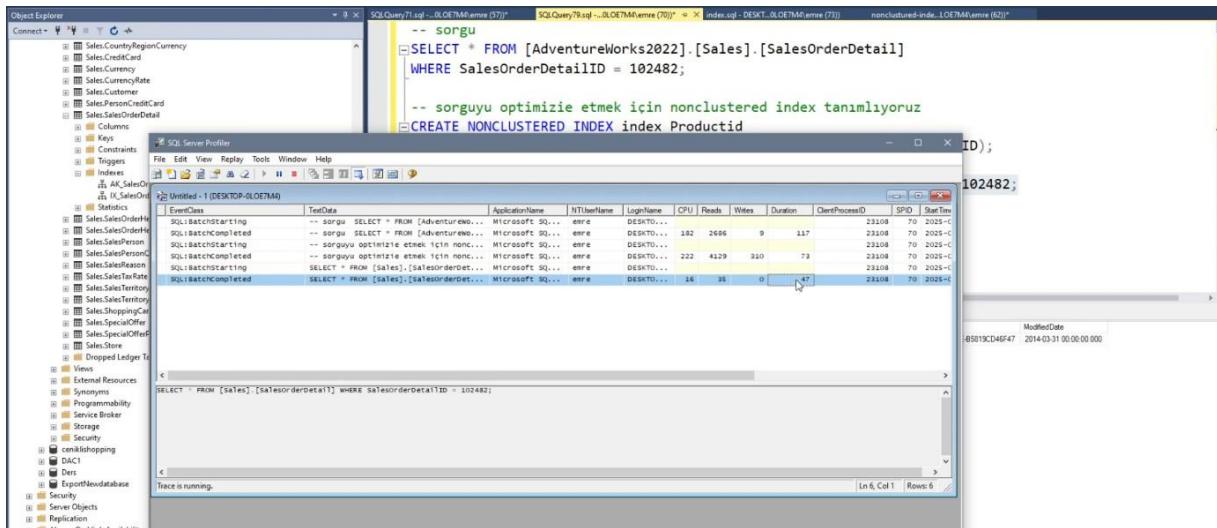
Şekil 9

Şekil 8'de kullandığımız tabloda ProductID üzerinde nonclustured index olmadan sorgu yaptığımızda 934 okuma yapıldığı ve duration olarak 55 zaman aldığı görülmektedir. Şekil 9'da ise ProductID için clustered index oluşturulduktan sonra 763 okuma yapılmış ve duration olarak 36 zaman aldığı görülmektedir. Yani clustered index oluşturulduktan sonra hem zaman olarak hem de okuma sayısı olarak sorgu daha performanslı çalışmıştır.

## c. Sorgu İyileştirme



Şekil 10



Şekil 11

Şekil 10'da kullandığımız tabloda SalesOrderDetailsID üzerinde nonclustured index olmadan sorgu yaptığımda 2686 okuma yapıldığı ve duration olarak 117 zaman aldığı görülmektedir. Şekil 11'de ise SalesOrderDetailsID için clustered index oluşturduktan sonra 35 okuma yapılmış ve duration olarak 47 zaman aldığı görülmektedir. Yani clustered index oluşturulduktan sonra hem zaman olarak hem de okuma sayısı olarak sorgu daha performanslı çalışmıştır.

Object Explorer

```
-- sorgu
SELECT * FROM [AdventureWorks2022].[Production].[WorkOrder]
WHERE ProductID = 3;

-- sorguyu optimizie etmek için nonclustered index tanımlıyoruz
CREATE NONCLUSTERED INDEX index_Productid
ON [AdventureWorks2022].[Production].[WorkOrder] (ProductID);

SELECT * FROM [AdventureWorks2022].[Production].[WorkOrder] WHERE ProductID= 3;
```

Results

WorkOrderID	ProductID	OrderQty	StockedQty	ShippedQty	StartDate	EndDate	DueDate	ShipReasonID	ModifiedDate
1	1	8	0	0	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
2	1	75	15	15	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
3	3	726	9	0	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
4	4	729	16	16	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
5	5	730	14	14	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
6	6	732	16	16	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
7	7	733	4	4	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
8	8	736	19	19	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
9	9	741	2	2	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
10	10	742	3	3	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
11	11	743	1	1	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
12	12	745	1	1	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
13	13	747	4	4	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
14	14	748	2	2	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
15	15	749	4	4	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000
16	16	753	14	14	2011-06-03 00:00:00.000	2011-06-13 00:00:00.000	2011-06-14 00:00:00.000	NULL	2011-06-11 00:00:00.000

Şekil 12

Object Explorer

```
-- sorgu
SELECT * FROM [AdventureWorks2022].[Production].[WorkOrder]
WHERE ProductID = 3;

-- sorguyu optimizie etmek için nonclustered index tanımlıyoruz
CREATE NONCLUSTERED INDEX index_Productid
ON [AdventureWorks2022].[Production].[WorkOrder] (ProductID);
```

SQL Server Profiler

ProductID = 3;

Events

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime
SQL:BatchStarting	-- sorgu SELECT * FROM [Adventureworks2022].[Production].[WorkOrder] WHERE ProductID = 3;	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				
SQL:BatchCompleted	-- sorgu SELECT * FROM [Adventureworks2022].[Production].[WorkOrder] WHERE ProductID = 3;	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				

Şekil 13

Object Explorer

```
-- sorgu
SELECT * FROM [AdventureWorks2022].[Production].[WorkOrder]
WHERE ProductID = 3;

-- sorguyu optimizie etmek için nonclustered index tanımlıyoruz
CREATE NONCLUSTERED INDEX index_Productid
ON [AdventureWorks2022].[Production].[WorkOrder] (ProductID);
```

SQL Server Profiler

ProductID = 3;

Events

EventClass	TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration	ClientProcessID	SPID	StartTime
SQL:BatchStarting	-- sorgu SELECT * FROM [Adventureworks2022].[Production].[WorkOrder] WHERE ProductID = 3;	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				
SQL:BatchCompleted	-- sorgu SELECT * FROM [Adventureworks2022].[Production].[WorkOrder] WHERE ProductID = 3;	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				
SQL:BatchStarting	-- sorguya optimize etmek için nonclustered index tanımlıyoruz	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				
SQL:BatchCompleted	-- sorguya optimize etmek için nonclustered index tanımlıyoruz	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				
SQL:BatchStarting	SELECT * FROM [Adventureworks2022].[Production].[WorkOrder] WHERE ProductID = 3;	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				
SQL:BatchCompleted	SELECT * FROM [Adventureworks2022].[Production].[WorkOrder] WHERE ProductID = 3;	Microsoft SQL Server	enre	DESKTOP-LOE7M4\enre	23108	57	2025-C				

Şekil 14

Şekil 12'de görülen sorgular ProductID üzerinde sırasıyla çalıştırılmıştır. Önce clustered index olmadan çalıştırılmış ve 1915 sorgu yapılmış, 231 duration zaman aldığı Şekil 13'te görülmektedir. Sonra clustered index oluşturulup aynı sorgu yapıldığında 553 sorgu yapılmış ve 91 duration zaman aldığı Şekil 14'te görülmektedir.

#### d. Veri Yöneticisi Rolleri

```
-- Sunucu düzeyinde login oluştur
CREATE LOGIN berkan WITH PASSWORD = 'berkan1234';

-- Veritabanı içinde kullanıcı oluştur
USE AdventureWorks2022;
CREATE USER berkan FOR LOGIN berkan;

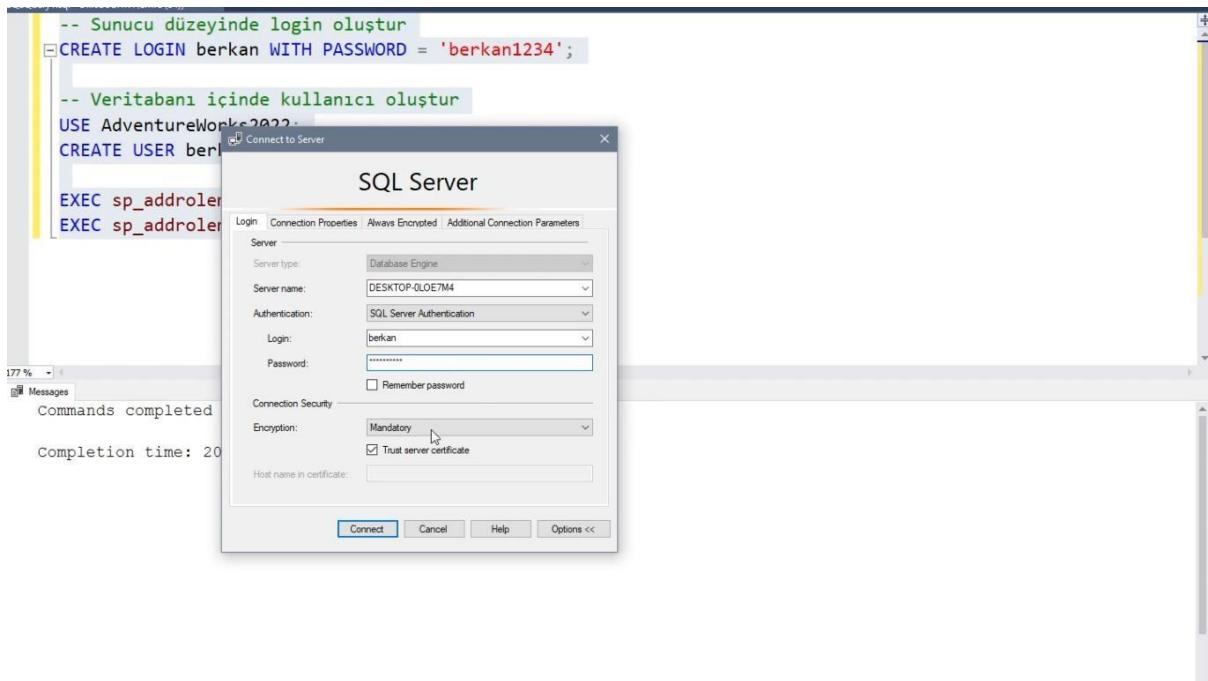
EXEC sp_addrolemember 'db_datareader', 'berkan';
EXEC sp_addrolemember 'db_denydatawriter', 'berkan';

Commands completed successfully.

Completion time: 2025-04-23T18:38:16.2613680+03:00
```

Şekil 15

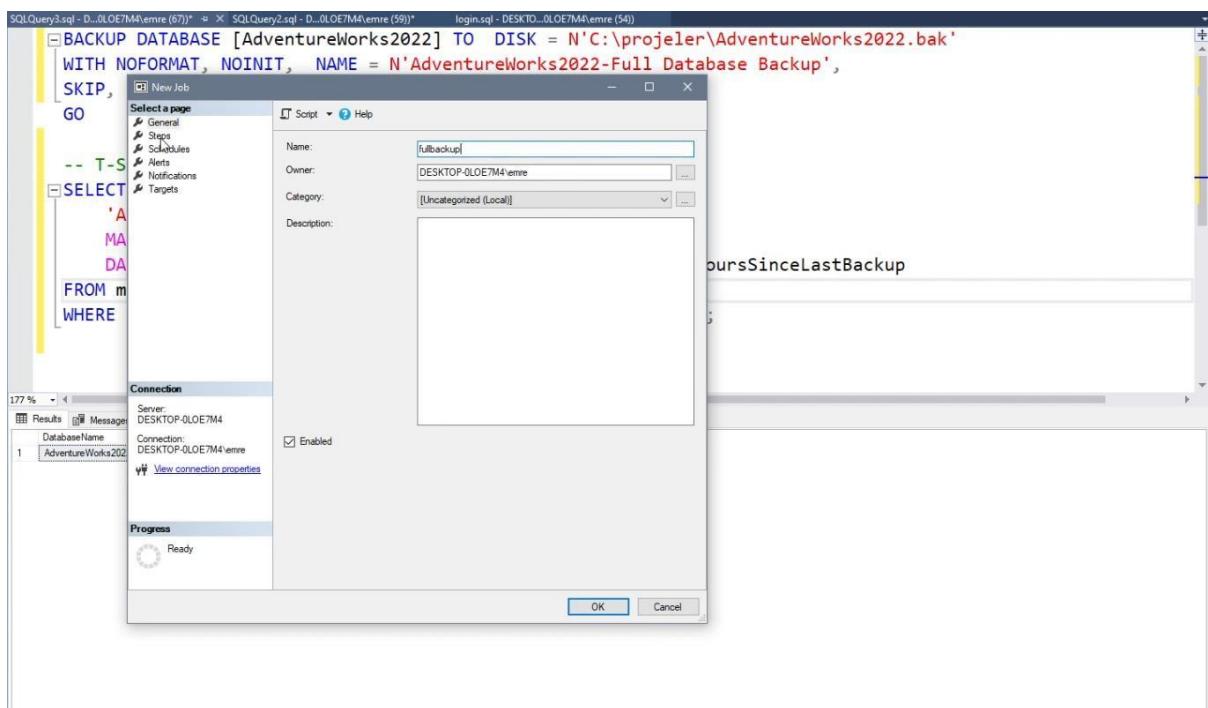
Şekil 15'te sunucu düzeyinde login oluşturmak için ve veritabanı içinde kullanıcı oluşturmak için yazılan sql kodları görülmektedir. Kodda bir kullanıcı adı ve parola atanmaktadır.



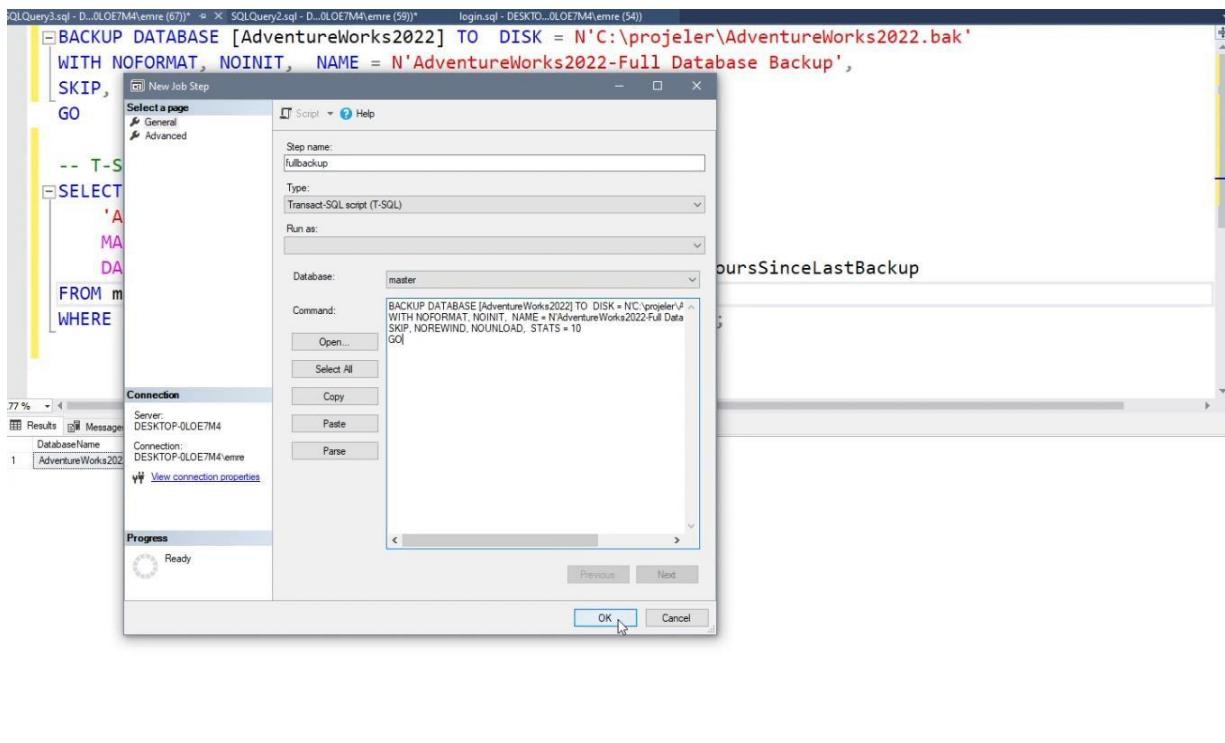
Şekil 16

Şekil 16'da sql servera bağlanırken Şekil 15'te atadığımız kullanıcı adı ve parolayı kullanıyoruz. Ardından veritabanında değişiklikler yapmak istediğimizde yapabiliyoruz.

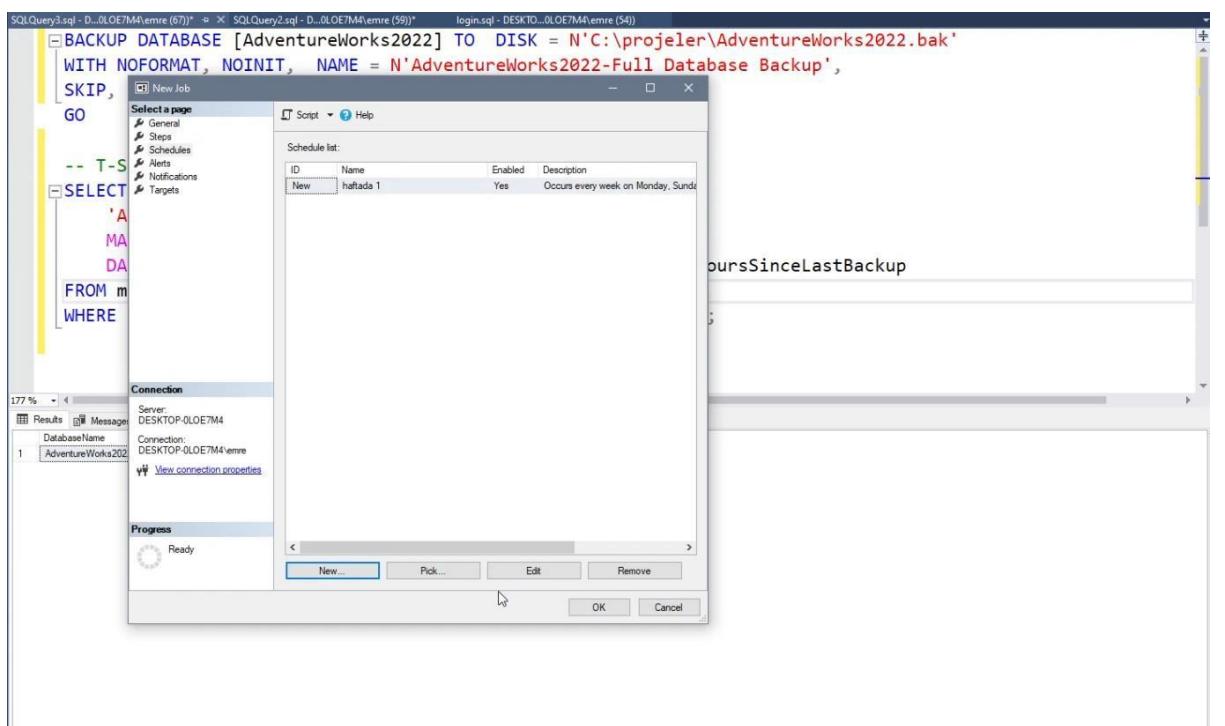
### 3. Veritabanı Yedekleme ve Otomasyon Çalışması



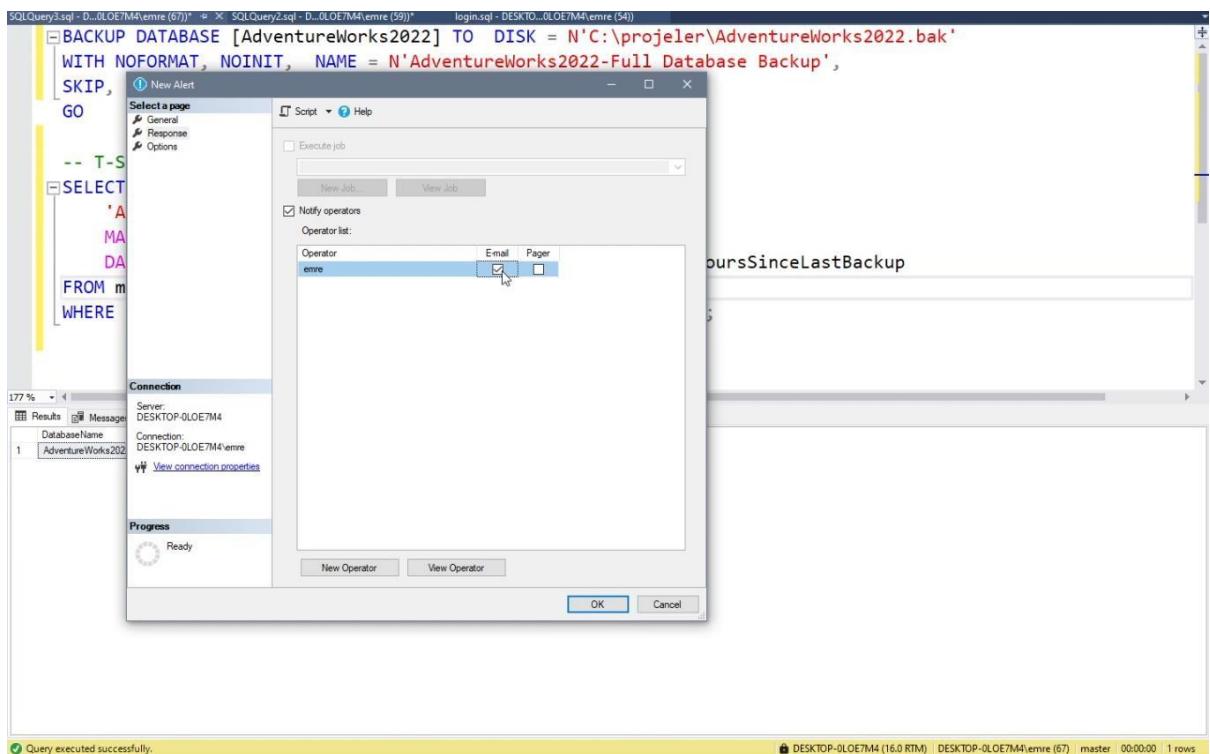
Şekil 17.1



Şekil 17.2



Şekil 17.3



Şekil 17.4

Şekil 17.1,17.2,17.3 ve 17.4'te bir sql agenti oluşturup yapılandırmalarının yapıldığı görülmektedir. 17.1'de ismi, 17.2 de agenta verilen komut, 17.3'te hangi aralıklarla yedekleme yapılacağı, 17.4'te ise raporun nasıl gönderileceğinin yapılandırmasının yapıldığı görülmektedir.

The screenshot shows a SQL Server Management Studio (SSMS) window with two tabs open: 'SQLQuery3.sql' and 'login.sql'. The 'SQLQuery3.sql' tab contains the following T-SQL script:

```
BACKUP DATABASE [AdventureWorks2022] TO DISK = N'C:\projeler\AdventureWorks2022.bak'
WITH NOFORMAT, NOINIT, NAME = N'AdventureWorks2022-Full Database Backup',
SKIP, NOREWIND, NOUNLOAD, STATS = 10
GO

-- T-SQL Scripting ile yedekleme raporları oluşturma.
SELECT
    'AdventureWorks2022' AS DatabaseName,
    MAX(b.backup_finish_date) AS LastBackupDate,
    DATEDIFF(HOUR, MAX(b.backup_finish_date), GETDATE()) AS HoursSinceLastBackup
FROM msdb.dbo.backupset b
WHERE b.database_name = 'AdventureWorks2022' AND b.type = 'D';
```

The 'Results' pane below the script displays the output of the SELECT query:

	DatabaseName	LastBackupDate	HoursSinceLastBackup
1	AdventureWorks2022	2025-04-24 02:18:00.000	0

Şekil 18

Şekil 18'de ise T-SQL Scripting ile yedekleme raporları oluşturmanın kodları verilmiştir.

## 4. Veritabanı Yedekleme ve Felaketten Kurtarma Planı

- Tam, Artık, Fark Yedeklemeleri:
  - a. Tam Yedekleme (Full Backup)

The screenshot shows a SQL query window with the following T-SQL code:

```
-- Full back up oluşturma
BACKUP DATABASE [AdventureWorks2022]
TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\FullBackUp.bak'
WITH NOFORMAT, NOINIT, NAME = N'AdventureWorks2022-Full Database Backup',
STATS = 10
GO
-- Full back up kontrol
RESTORE HEADERONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\FullBackUp.bak'
GO
```

Below the code, the 'Messages' tab displays the progress of the backup operation:

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 25680 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.  
Processed 1 pages for database 'AdventureWorks2022', file 'AdventureWorks2022\_log' on file 1.  
BACKUP DATABASE successfully processed 25681 pages in 0.649 seconds (309.131 MB/sec).

Şekil 19

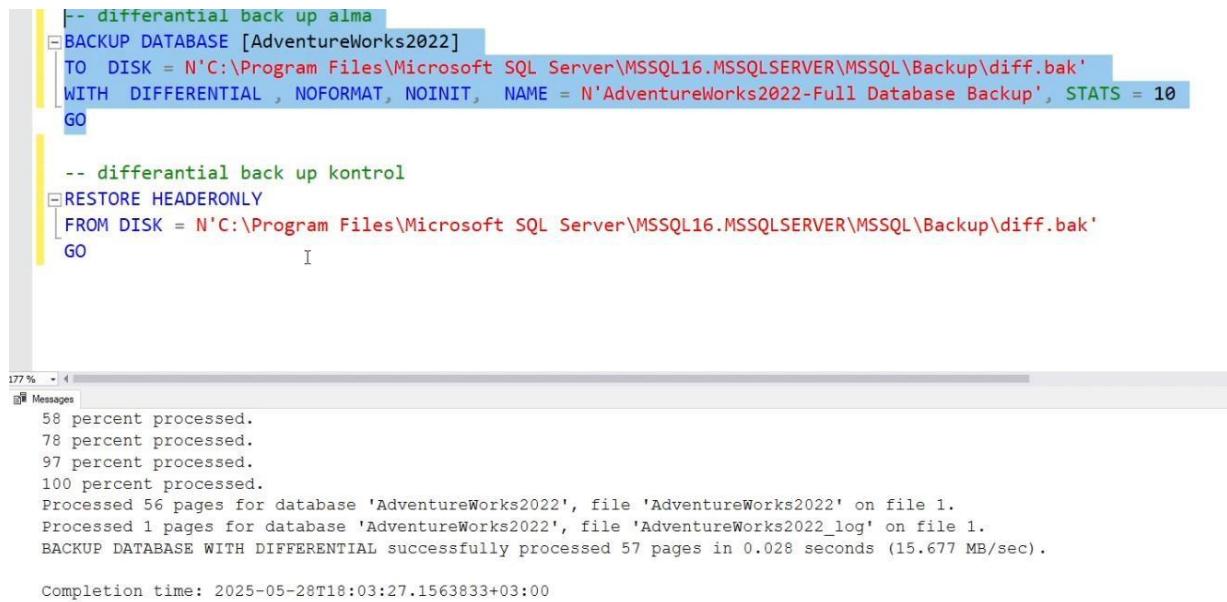
The screenshot shows a SQL query window with the same T-SQL code as in Figure 19. Below the code, the 'Results' tab displays a grid of backup information:

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate	BackupSize	FirstLsn
AdventureWorks2022-Full Database Backup	NULL	1	NULL	0	1	2	DESKTOP-0LOE7M4\emre	DESKTOP-0LOE7M4	AdventureWorks2022	957	2025-04-24 14:20:25.000	211898368	58

Şekil 20

Bir veri tabanının veya dosya sisteminin tüm verilerinin bir kopyasının alındığı yedekleme türüdür. Şekil 19 ve Şekil 20 de yedekleme kodu ve yedeklemeyi kontrol etme kodu verilmiştir.

## b. Fark Yedekleme (Diff Backup)



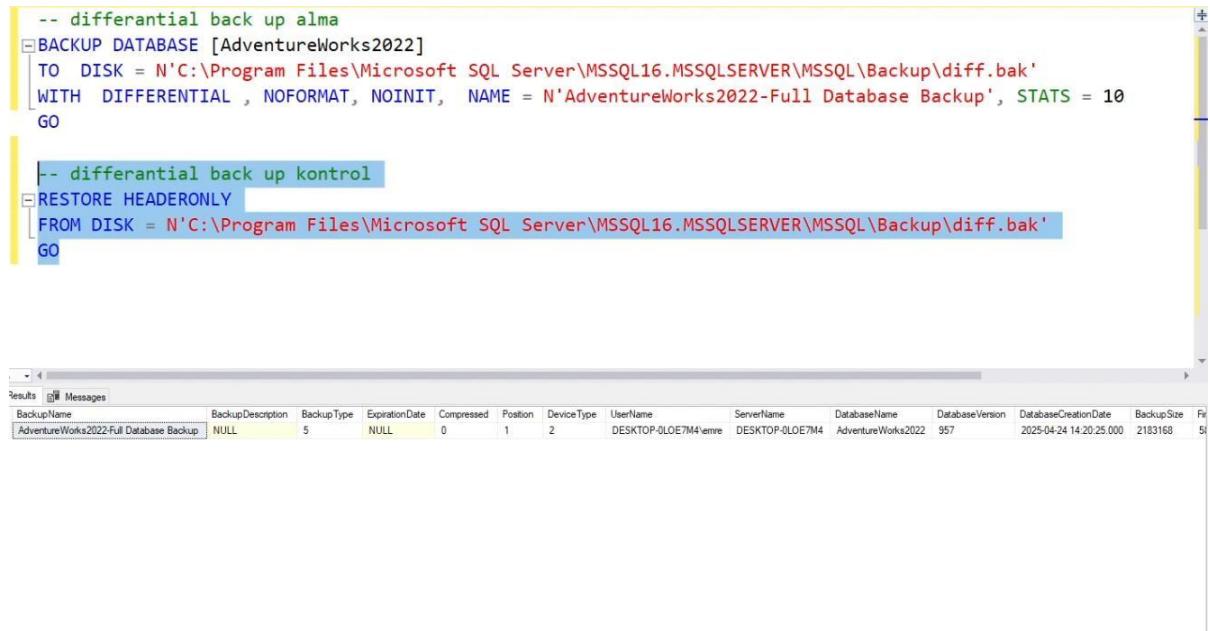
```
-- differential back up alma
BACKUP DATABASE [AdventureWorks2022]
TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\diff.bak'
WITH DIFFERENTIAL , NOFORMAT, NOINIT, NAME = N'AdventureWorks2022-Full Database Backup', STATS = 10
GO

-- differential back up kontrol
RESTORE HEADERONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\diff.bak'
GO
```

58 percent processed.  
78 percent processed.  
97 percent processed.  
100 percent processed.  
Processed 56 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.  
Processed 1 pages for database 'AdventureWorks2022', file 'AdventureWorks2022\_log' on file 1.  
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 57 pages in 0.028 seconds (15.677 MB/sec).

Completion time: 2025-05-28T18:03:27.1563833+03:00

Şekil 21



```
-- differential back up alma
BACKUP DATABASE [AdventureWorks2022]
TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\diff.bak'
WITH DIFFERENTIAL , NOFORMAT, NOINIT, NAME = N'AdventureWorks2022-Full Database Backup', STATS = 10
GO

-- differential back up kontrol
RESTORE HEADERONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\diff.bak'
GO
```

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate	BackupSize	File
AdventureWorks2022-Full Database Backup	NULL	5	NULL	0	1	2	DESKTOP-0LOE7M4\enrre	DESKTOP-0LOE7M4	AdventureWorks2022	957	2025-04-24 14:20:25,000	2183168	51

Şekil 22

Fark yedekleme, son tam yedeklemeden bu yana değişen tüm verileri yedekler. Örneğin Pazartesi günü tam yedek alındıysa, Çarşamba günü diffbackup hem Salı hem Çarşamba yapılan değişiklikleri içerir. Şekil 21 ve 22 de diffbackup alma ve kontrol kodu gösterilmiştir.

## c. Transaction Log Backup

```
-- transaction log back up alma
BACKUP LOG [AdventureWorks2022]
TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\Transaction.bak'
WITH NOFORMAT, NOINIT, NAME = N'AdventureWorks2022-Full Database Backup', STATS = 10
GO

-- transaction back up kontrol
RESTORE HEADERONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\Transaction.bak'
GO
```

100 percent processed.  
Processed 3 pages for database 'AdventureWorks2022', file 'AdventureWorks2022\_log' on file 1.  
BACKUP LOG successfully processed 3 pages in 0.015 seconds (1.106 MB/sec).  
Completion time: 2025-05-28T18:36:07.3268477+03:00

Şekil 23

```
-- transaction log back up alma
BACKUP LOG [AdventureWorks2022]
TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\Transaction.bak'
WITH NOFORMAT, NOINIT, NAME = N'AdventureWorks2022-Full Database Backup', STATS = 10
GO

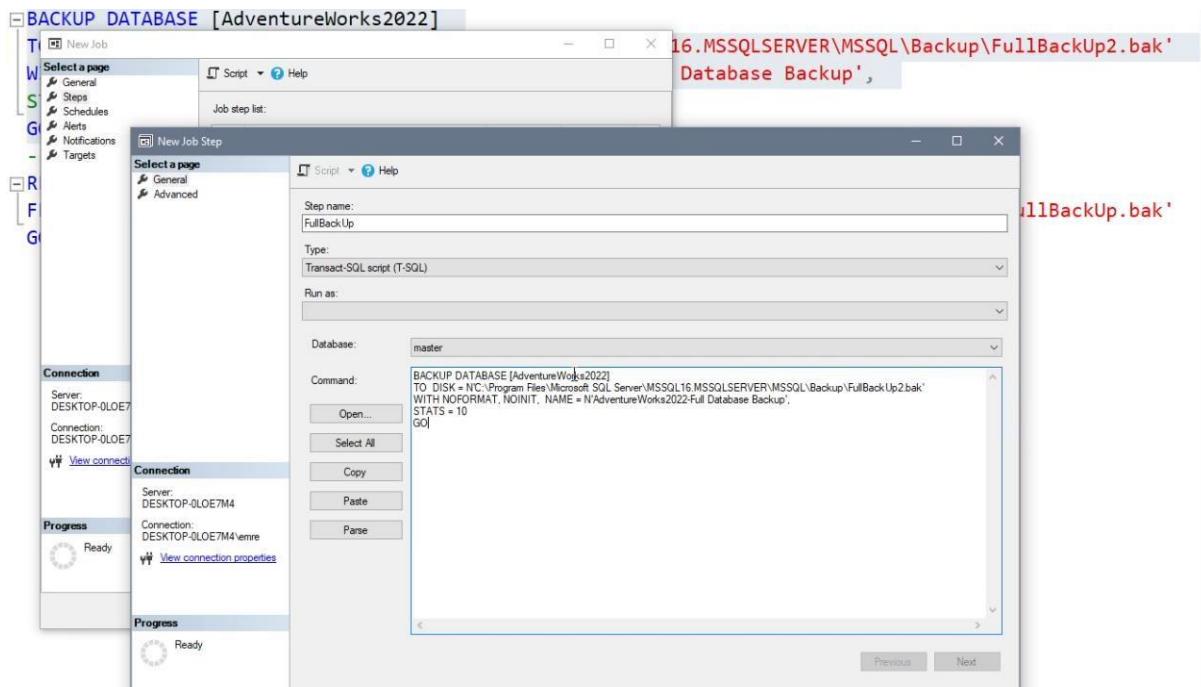
-- transaction back up kontrol
RESTORE HEADERONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\Transaction.bak'
GO
```

BackupName	BackupDescription	BackupType	ExpirationDate	Compressed	Position	DeviceType	UserName	ServerName	DatabaseName	DatabaseVersion	DatabaseCreationDate	BackupSize	FirstLSN
AdventureWorks2022-Full Database Backup	NULL	2	NULL	0	1	2	DESKTOP-0LOE7M4\enre	DESKTOP-0LOE7M4	AdventureWorks2022	957	2025-04-24 14:20:25.000	81920	5800000

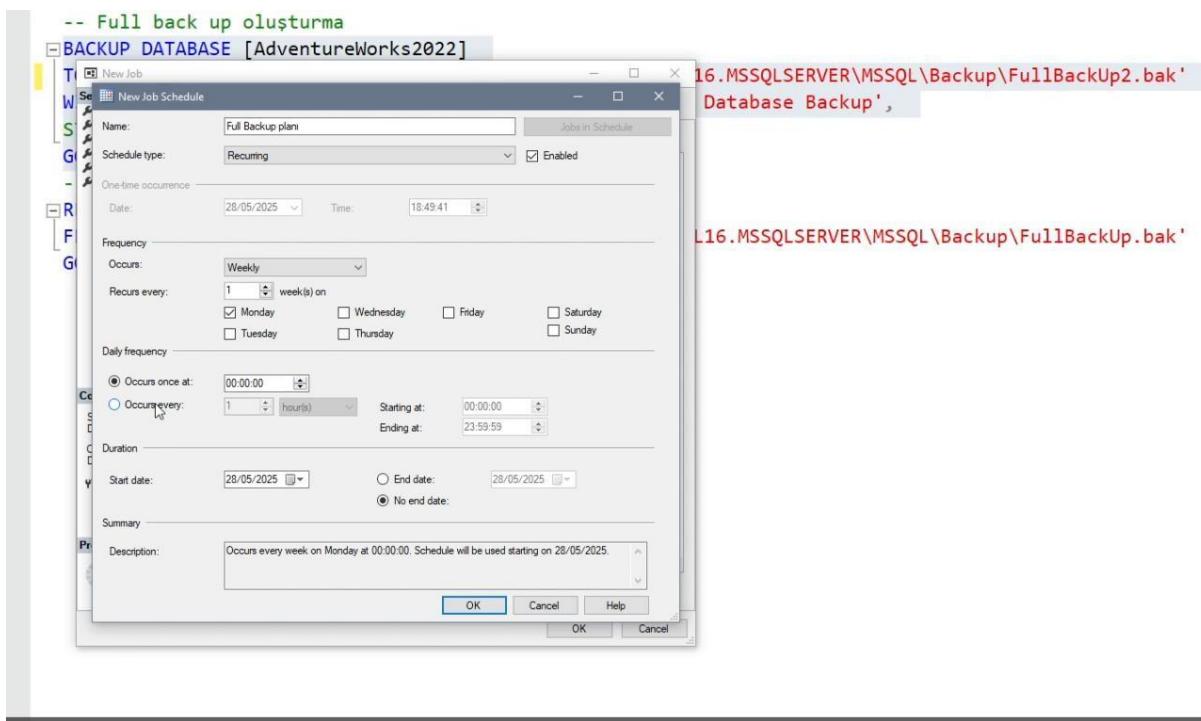
Şekil 24

Bir veritabanı sisteminde gerçekleşen tüm işlemlerin (transaction) kayıt altına alındığı işlem günlüklerinin yedeğini almak için kullanılan bir yedekleme türüdür. Bu yedekleme türü, özellikle veri kaybını en aza indirmek ve istenen herhangi bir zamana kadar geri dönüş (point-in-time recovery) yapılabilmesini sağlamak amacıyla kullanılır. Tam (full) veya fark (differential) yedekler sadece veritabanının belirli anlık durumlarını saklarken, transaction log yedekleri bu anlar arasında gerçekleşen tüm değişiklikleri ayrıntılı biçimde kaydeder. Şekil 23 ve Şekil 24'te transaction log backup alma ve kontrol etme kodu gösterilmiştir.

## • Zamanlayıcılarla Yedekleme



Şekil 25



Şekil 26

Şekil 25 ve 26'da zamanlayıcı ile yedekleme yöntemi gösterilmiştir. Yeni bir iş oluşturma kısmına gelip önce iş için full back up sql kodunu giriyoruz ardından zamanlamasını seçiyoruz ve ayarlıyoruz.

## • Felaketten Kurtarma Senaryaları

```
-- En sonki ana dönmek için Transaction log back up geri yükleme işlemi. Bunun için FullBackup.bak ve diff.bak da gereklidir
USE master
RESTORE DATABASE [AdventureWorks2022]
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\FullBackup.bak'
WITH FILE = 1, NORECOVERY, NOUNLOAD, REPLACE, STATS = 5

RESTORE DATABASE [AdventureWorks2022]
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\diff.bak'
WITH FILE = 1, NORECOVERY, NOUNLOAD, STATS = 5

RESTORE LOG [AdventureWorks2022]
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\Transaction.bak'
WITH FILE = 1, NOUNLOAD, STATS = 5
GO
```

Şekil 27

Şekil 27'de felaketten kurtarma senaryosu gösterilmiştir. Veriler silindikten sonra, önce full backup sonra diff backup ardından transaction log backup yüklenmiş ve silinmeden önceki son duruma dönülmüştür.

## • Test Yedekleme Senaryaları

```
--Yedeğin bozuk olup olmadığını kontrol eder (geri yüklemeden).
RESTORE VERIFYONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\FullBackup.bak'
RESTORE VERIFYONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\diff.bak'
RESTORE VERIFYONLY
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\Transaction.bak'

-- Manuel olarak geri yükleme kontrol testi
USE [AdventureWorks2022]
GO
SELECT *
FROM [AdventureWorks2022].[Sales].[CurrencyRate] where ToCurrencyCode = 'BRL'

DELETE FROM [Sales].[CurrencyRate]
WHERE ToCurrencyCode = 'BRL'

USE [master]
RESTORE DATABASE [AdventureWorks2022]

Messages
The backup set on file 1 is valid.
The backup set on file 1 is valid.
The backup set on file 1 is valid. I

Completion time: 2025-05-28T20:20:21.6722817+03:00
```

Şekil 28

Şekil 28'de yedeğin bozuk olup olmadığını geri yüklemeden kontrol etme kodu gösterilmiştir.

```
USE [master]
RESTORE DATABASE [AdventureWorks2022]
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\FullBackUp.bak'
WITH FILE = 1, NORECOVERY, NOUNLOAD, REPLACE, STATS = 5
RESTORE DATABASE [AdventureWorks2022]
FROM DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\diff.bak'
WITH FILE = 1, RECOVERY, NOUNLOAD, STATS = 5
GO

6 Messages
95 percent processed.
100 percent processed.
Processed 25680 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.
Processed 1 pages for database 'AdventureWorks2022', file 'AdventureWorks2022_log' on file 1.
RESTORE DATABASE successfully processed 25681 pages in 0.357 seconds (561.978 MB/sec).
100 percent processed.
Processed 64 pages for database 'AdventureWorks2022', file 'AdventureWorks2022' on file 1.
Processed 1 pages for database 'AdventureWorks2022', file 'AdventureWorks2022_log' on file 1
```

Şekil 29

Şekil 29'da ise önce veriler silinmiş sonra geri yüklenerek manuel olarak kontrol edilerek test edilmiştir.

## 5. Veritabanı Güvenliği

### a. Erişim Yönetimi

```
SELECT CASE SERVERPROPERTY('IsIntegratedSecurityOnly')
WHEN 1 THEN 'Windows Authentication Mode'
WHEN 0 THEN 'Mixed Mode'
END as [Authentication Mode]

--SQL Login oluştur
-- Veritabanı kullanıcısı ve yetkiler
CREATE LOGIN user2 WITH PASSWORD = 'Password123';
CREATE USER user2 FOR LOGIN user2;
ALTER ROLE db_datareader ADD MEMBER user2; -- sadece okuma yetkisi

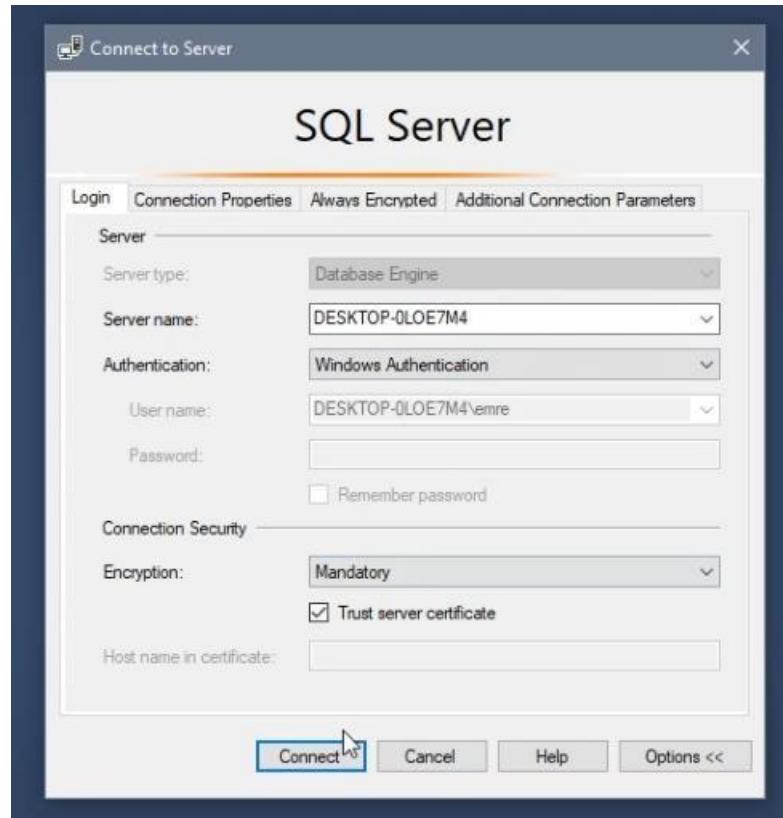
-- izinleri test etme
SELECT * FROM [AdventureWorks2022].[Sales].[Currency]; -- Başarılı
DELETE FROM Sales.SalesOrderHeader; -- Yetki hatalı

7 % Messages
(0 rows affected)

Completion time: 2025-05-29T15:26:56.5244718+03:00
```

Şekil 30

Şekil 30'da SQL Server Authentication ile erişim yönetimi oluşturulmuştur. Kullanıcıya sadece okuma yetkisi verilmiştir. Kullanıcı silme işlemi yapmak istediği yapılamadığı gösterilmiştir.



Şekil 31

Şekil 31'de Windows Authentication ile erişim gösterilmiştir.

## b. Veri şifreleme

```
-- 1. Master Key oluştur
USE AdventureWorks2022;
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'StrongMasterKey@123';

-- 2. Sertifika oluştur
CREATE CERTIFICATE TdeCert WITH SUBJECT = 'TDE Certificate';

-- 3. Database Encryption Key oluştur
USE AdventureWorks2022;
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE TdeCert;

-- 4. Şifrelemeyi etkinleştir
ALTER DATABASE AdventureWorks2022 SET ENCRYPTION ON;
```

Şekil 32

```
-- Tüm veritabanlarının şifreleme durumu
SELECT
    db.name,
    dek.encryption_state,
    CASE dek.encryption_state
        WHEN 0 THEN 'No encryption'
        WHEN 1 THEN 'Unencrypted'
        WHEN 2 THEN 'Encryption in progress'
        WHEN 3 THEN 'Encrypted'
        WHEN 4 THEN 'Key change in progress'
        WHEN 5 THEN 'Decryption in progress'
        WHEN 6 THEN 'Protection change in progress'
    END AS encryption_state_desc,
    dek.percent_complete,
    dek.key_algorithm,
    dek.key_length
FROM sys.dm_database_encryption_keys dek
JOIN sys.databases db ON dek.database_id = db.database_id;
```

77 % ▾

Results Messages

	name	encryption_state	encryption_state_desc	percent_complete	key_algorithm	key_length
1	tempdb	3	Encrypted	0	AES	256
2	ceniklisshopping	1	Unencrypted	0	AES	256
3	AdventureWorks2022	2	Encryption in progress	100	AES	256

Şekil 33

Şekil 32 ve 33'te veri şifreleme gösterilmiştir. Önce masterkey oluşturulmuş sonra encryption key ve sertfika oluşturulmuş ardından şifreleme etkinleştirilmiştir. Bu adımlar TDE - Transparent Data Encryption'in temel adımlarıdır. Daha sonra tüm veritabanı şifrelenmiştir.

## c. SQL Injection Testleri

```
--1. Test Amaçlı Sakıncalı (Injection'a açık) Stored Procedure Oluştur
USE AdventureWorks2022;
GO

CREATE PROCEDURE dbo.usp_InsecureEmployeeSearch
    @LastName NVARCHAR(100)
AS
BEGIN
    DECLARE @sql NVARCHAR(MAX);
    SET @sql = 'SELECT BusinessEntityID, FirstName, LastName
                FROM Person.Person
                WHERE LastName = ''' + @LastName + '''';

    EXEC sp_executesql @sql;
END;
GO

--Normal Kullanım:
EXEC dbo.usp_InsecureEmployeeSearch @LastName = 'Smith';

--SQL Injection Saldırısı: "OR 1=1" sayesinde tüm çalışanları döndürür. Bu klasik bir SQL injection örneğidir.
EXEC dbo.usp_InsecureEmployeeSearch @LastName = 'Smith'' OR 1=1 --';
```

Şekil 34

```
--Güvenli Versiyon: Parametreli Sorgu Kullan
USE AdventureWorks2022;
GO

CREATE PROCEDURE dbo.usp_SecureEmployeeSearch
    @LastName NVARCHAR(100)
AS
BEGIN
    SELECT BusinessEntityID, FirstName, LastName
    FROM Person.Person
    WHERE LastName = @LastName;
END;
GO

--Güvenli kullanım: Parametre doğrudan SQL'e gömülmediği için injection burada işe yaramaz.
EXEC dbo.usp_SecureEmployeeSearch @LastName = 'Smith';
EXEC dbo.usp_SecureEmployeeSearch @LastName = 'Smith'' OR 1=1 --';
```

Şekil 35

Şekil 34'te test amaçlı injectiona açık parametrenin doğrudan SQL'e gömüldüğü stored procedure oluşturulmuş ve lastname ='smith' or 1=1 – ' diye sorgu yapıldığında tüm çalışan listesi gelmiştir fakat parametrenin doğrudan SQL'e gömülmediği durumda injection işe yaramamıştır.

## d. Audit Logları

```
USE master;
GO
--Audit Loglarını Saklamak için Audit Nesnesi Oluştur
CREATE SERVER AUDIT AuditAdventureWorks
TO FILE (
    FILEPATH = 'C:\AuditLogs2\', -- Bu klasör gerçekten var olmalı
    MAXSIZE = 10 MB,
    MAX_FILES = 5,
    RESERVE_DISK_SPACE = OFF
)
WITH (
    QUEUE_DELAY = 1000,
    ON_FAILURE = CONTINUE
);
GO
--Veritabanı Üzerinde Audit Specification Oluştur
USE AdventureWorks2022;
GO
```

Şekil 36

```
--Veritabanı Üzerinde Audit Specification Oluştur
USE AdventureWorks2022;
GO

CREATE DATABASE AUDIT SPECIFICATION Audit_Read_Operations
FOR SERVER AUDIT AuditAdventureWorks
ADD (SELECT ON SCHEMA::HumanResources BY public), -- Tüm kullanıcılar için SELECT
ADD (INSERT ON SCHEMA::HumanResources BY public), -- Tüm kullanıcılar için INSERT
ADD (DELETE ON SCHEMA::HumanResources BY public) -- Tüm kullanıcılar için DELETE
WITH (STATE = ON);
GO
-- Audit'i Aktif Et
use master
ALTER SERVER AUDIT AuditAdventureWorks WITH (STATE = ON);

-- Örneğin bir SELECT işlemi yap
USE AdventureWorks2022;
SELECT * FROM HumanResources.Employee;

-- INSERT denemesi
INSERT INTO HumanResources.EmployeeDepartmentHistory
(BusinessEntityID, DepartmentID, ShiftID, StartDate)
VALUES (1, 3, 2, GETDATE());
```

Şekil 37

```
-- Logları Görüntülemek
SELECT *
FROM sys.fn_get_audit_file('C:\AuditLogs2\AuditAdventureWorks*.sqlaudit', DEFAULT, DEFAULT);
```

The screenshot shows the SQL Server Management Studio interface with the 'Messages' tab selected. The results pane displays the output of the audit log query. The data is presented in a table with columns: ver\_time, sequence\_number, action\_id, succeeded, permission\_bmask, is\_column\_permission, session\_id, server\_principal\_id, database\_principal\_id, target\_server\_principal\_id, and target\_database\_principal\_id. The data shows three rows of audit events: one AUSC event at 14:41:55.0799170, one SL event at 14:41:58.8856715, and one IN event at 14:42:07.4526891.

ver_time	sequence_number	action_id	succeeded	permission_bmask	is_column_permission	session_id	server_principal_id	database_principal_id	target_server_principal_id	target_database_principal_id
1025-05-29 14:41:55.0799170	1	AUSC	1	0x00	0	59	259	0	0	0
1025-05-29 14:41:58.8856715	1	SL	1	0x0001	1	59	259	1	0	0
1025-05-29 14:42:07.4526891	1	IN	1	0x0008	0	59	259	1	0	0

Şekil 38

Şekil 36'da önce bir audit nesnesi oluşturulmuş, kaydedileceği dosyanın yolu verilmiştir. Daha sonra Şekil 37'de audit specification oluşturulmuş yetkiler belirlenmiştir ve bir select işlemi bir de insert işlemi yapılmıştır (bir okuma bir yazma işlemi). Sonrasında şekil 38'de logları görüntüleme kodu çalıştırıldığında hareketler izlenmiştir.

## 6. Veritabanı Yükseltme ve Sürüm Yönetimi

### a. Veritabanı Yükseltme Planı

The screenshot shows the SQL Server Management Studio interface with the 'Results' tab selected. The results pane displays the execution of a script that creates a 'DatabaseVersion' table and a 'Customers' table, and inserts initial data into the 'DatabaseVersion' table. The final command executed is 'SELECT \* FROM DatabaseVersion'.

```

CREATE TABLE DatabaseVersion (
    VersionID INT IDENTITY(1,1) PRIMARY KEY,
    VersionNumber VARCHAR(20),
    Description NVARCHAR(500),
    Status VARCHAR(50),
    AppliedDate DATETIME DEFAULT GETDATE()
);

-- İlk versiyon kaydı (başlangıç)
INSERT INTO DatabaseVersion (VersionNumber, Description, Status)
VALUES ('1.0.0', 'Initial setup - version system created', 'Completed');

CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY IDENTITY(1,1),
    FirstName NVARCHAR(50),
    LastName NVARCHAR(50),
    Email NVARCHAR(100)
);

-- INSERT INTO DatabaseVersion (VersionNumber, Description, Status)
-- VALUES ('1.1.0', 'Created Customers table', 'Completed');

SELECT * FROM DatabaseVersion

```

Results

VersionID	VersionNumber	Description	Status	AppliedDate
1	1.0.0	Initial setup - version system created	Completed	2025-05-30 06:29:20.070
2	1.1.0	Created Customers table	Completed	2025-05-30 06:31:00.977

Şekil 39

Şekil 39'da versiyon değişikliklerini göstermek için gereken tablo kurulmuş ve örnek bir değişiklik yapılmıştır.

```
SELECT * FROM SchemaChanges

CREATE TRIGGER trg_SchemaChangeLogger
ON DATABASE
FOR CREATE_TABLE, ALTER_TABLE, DROP_TABLE
AS
BEGIN
    DECLARE @data XML = EVENTDATA();
    INSERT INTO SchemaChanges (
        EventType,
        ObjectName,
        SqlCommand,
        PerformedBy
    )
    VALUES (
        @data.value('/EVENT_INSTANCE/EventType[1]', 'NVARCHAR(100)'),
        @data.value('/EVENT_INSTANCE/ObjectName[1]', 'NVARCHAR(100)'),
        @data.value('/EVENT_INSTANCE/TSQLCommand[1]', 'NVARCHAR(MAX)'),
        SYSTEM_USER
    );
END;

CREATE TABLE TestLogTable2 (
    ID INT PRIMARY KEY,
    Name NVARCHAR(50)
);

SELECT * FROM SchemaChanges ORDER BY ChangeDate DESC;
```

The screenshot shows the SQL Server Management Studio interface. The top pane displays the T-SQL script for creating a trigger named 'trg\_SchemaChangeLogger' on the database level. This trigger logs events for CREATE\_TABLE, ALTER\_TABLE, and DROP\_TABLE operations into a table named 'SchemaChanges'. The 'SchemaChanges' table has columns for EventType (NVARCHAR(100)), ObjectName (NVARCHAR(100)), SqlCommand (NVARCHAR(MAX)), and PerformedBy (SYSTEM\_USER). The bottom pane shows the results of the 'SELECT \* FROM SchemaChanges ORDER BY ChangeDate DESC;' query, which returns a single row with the following data:

	ChangeID	EventType	ObjectName	SqlCommand	ChangeDate	PerformedBy
1	1	CREATE_TABLE	TestLogTable	CREATE TABLE TestLogTable2 ( ID INT PRIMARY KEY, Name NVARCHAR(50) );	2025-05-30 06:37:05.287	berkanmsilberkan

Şekil 40

Şekil 40'ta DDL Triggers ile yapılan değişiklikleri otomatik kaydeden trigger sistemi oluşturulmuştur, yapılan örnek işleminden sonra işlemin trigger tablosuna kaydedildiği görülmektedir.

- Versiyon Yükseltme Prosedürü

```

CREATE PROCEDURE pr_ApplyVersion
    @NewVersion VARCHAR(20),
    @Description NVARCHAR(500)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRAN;

        -- En son versiyonu al
        DECLARE @Current VARCHAR(20);
        SELECT TOP 1 @Current = VersionNumber
        FROM DatabaseVersion
        ORDER BY VersionID DESC;

        -- Yeni versiyon daha büyük olmalı
        IF @Current IS NOT NULL AND @Current >= @NewVersion
        BEGIN
            RAISERROR('Bu versiyon zaten uygulanmış veya daha yeni.', 16, 1);
            ROLLBACK TRAN;
            RETURN;
        END

        -- Versiyon kaydını geçici olarak ekle
        INSERT INTO DatabaseVersion (VersionNumber, Description, Status)
        VALUES (@NewVersion, @Description, 'In Progress');

        -- !!! Buraya versiyon değişikliği yapılacak SQL komutları yazılır !!!

        -- Örnek: ALTER TABLE, CREATE TABLE vs.
        -- Örnek alanı boş bırakıldı, dışarıdan çalıştırılır.

        -- Başarılıysa statüyü güncelle
        UPDATE DatabaseVersion

```

Şekil 41

```

-- Başarılıysa statüyü güncelle
UPDATE DatabaseVersion
SET Status = 'Completed'
WHERE VersionNumber = @NewVersion;

COMMIT TRAN;
END TRY
BEGIN CATCH
    IF @@TRANCOUNT > 0
        ROLLBACK TRAN;

    -- Hata durumunda status'u güncelle
    UPDATE DatabaseVersion
    SET Status = 'Failed - ' + ERROR_MESSAGE()
    WHERE VersionNumber = @NewVersion;

    RAISERROR('Versiyon uygulama başarısız oldu.', 16, 1);
END CATCH
END;

```

Şekil 42

Şekil 41 ve 42' pr\_ApplyVersion prosedürü gösterilmiştir, bu prosedür veritabanında yapılan yapısal değişikliklerin belirli bir versiyon numarası ile kayıt altına alınmasını sağlayan prosedürdür. Yeni bir versiyon uygulanmadan önce mevcut versiyon kontrol edilir. Uygun ise versiyon bilgisi DatabaseVersion tablosuna "In Progress" olarak eklenir. İşlem başarılı olursa "Completed", hata oluşursa "Failed - [hata mesajı]" olarak güncellenir. Bu prosedür, veritabanı güncellemelerinin kontrollü ve izlenebilir şekilde yapılmasını sağlar.

- Geri Alma Prosedürü

```
CREATE PROCEDURE pr_RollbackVersion
    @TargetVersion VARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        -- Mevcut versiyonu al
        DECLARE @CurrentVersion VARCHAR(20);
        SELECT TOP 1 @CurrentVersion = VersionNumber
        FROM DatabaseVersion
        ORDER BY VersionID DESC;

        -- Geçerli versiyon, hedef versiyondan küçükse rollback yapılamaz
        IF @CurrentVersion <= @TargetVersion
        BEGIN
            RAISERROR('Rollback versiyonu mevcut versiyondan büyük veya eşit olamaz.', 16, 1);
            ROLLBACK TRANSACTION;
            RETURN;
        END

        -- Yeni rollback kaydı gir (şimdilik In Progress)
        INSERT INTO DatabaseVersion (VersionNumber, Description, Status)
        VALUES (@TargetVersion, 'Rollback to version ' + @TargetVersion, 'In Progress');
        -- !!! Buraya geri alma (rollback) işlemleri yazılır !!!
        -- Başarıyla tamamlandıysa status'u güncelle
        UPDATE DatabaseVersion
        SET Status = 'Reverted'
        WHERE VersionNumber = @TargetVersion;

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;

        UPDATE DatabaseVersion
        SET Status = 'Rollback Failed: ' + ERROR_MESSAGE()
        WHERE VersionNumber = @TargetVersion;

        RAISERROR('Geri alma işlemi başarısız oldu.', 16, 1);
    END CATCH

```

Şekil 43

- Şekil 43'te pr\_RollbackVersion isimli prosedür gösterilmiştir, bu prosedür veritabanını daha önceki bir versiyon durumuna geri almak amacıyla kullanılan prosedürdür. Hedef versiyon, mevcut versiyondan eski olmalıdır. Geri alma işlemi TRANSACTION bloğu içinde gerçekleştirilir. İşlem başarılı olursa versiyon bilgisi DatabaseVersion tablosuna "Reverted" olarak kaydedilir. Hata durumunda işlem geri alınır ve ilgili satır "Rollback Failed: [hata mesajı]" olarak güncellenir. Bu prosedür, istenmeyen değişikliklerin geri alınabilmesini ve sistemin güvenliğini sağlar.

```

CREATE PROCEDURE pr_ListVersions
AS
BEGIN
    SELECT * FROM DatabaseVersion ORDER BY VersionID DESC;
END;

```

Şekil 44

Şekil 44'te versiyonları kontrol etmek için gerekli olan prosedür gösterilmiştir.

- Test

```

-- TRANSACTION KULLANMA
CREATE TABLE Customers3 (
    CustomerID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName NVARCHAR(50),
    LastName NVARCHAR(50),
    Email NVARCHAR(100)
);

EXEC pr_ApplyVersion '1.2.0', 'Customers1 tablosu oluşturuldu.';

-- 1. Sütun varsa ekleme, yoksa ALTER TABLE çalıştır
IF NOT EXISTS (
    SELECT * FROM sys.columns
    WHERE object_id = OBJECT_ID('Customers1') AND name = 'Phone'
)
BEGIN
    ALTER TABLE Customers1 ADD Phone NVARCHAR(20);
END

-- 2. Versiyon kaydını ekle
EXEC pr_ApplyVersion '1.2.1', 'Customers1 tablosuna Phone sütunu eklendi.';

```

Results Messages

VersionID	VersionNumber	Description	Status	AppliedDate
1	5	Customers1 tablosu oluşturuldu.	Completed	2025-05-30 08:06:46.260
2	1.1.0	Created Customers table	Completed	2025-05-30 06:31:00.977
3	1.0.0	Initial setup - version system created	Completed	2025-05-30 06:29:20.070

Şekil 45

Şekil 45'te test etmek için yeni bir tablo oluşturduğumuz prosedürle eklenmiştir.

```

-- 1. Sütun varsa ekleme, yoksa ALTER TABLE çalıştır
IF NOT EXISTS (
    SELECT * FROM sys.columns
    WHERE object_id = OBJECT_ID('Customers3') AND name = 'Phone'
)
BEGIN
    ALTER TABLE Customers3 ADD Phone NVARCHAR(20);
END

-- 2. Versiyon kaydını ekle
EXEC pr_ApplyVersion '1.2.1', 'Customers1 tablosuna Phone sütunu eklendi.';

EXEC pr_ApplyVersion '1.2.1', 'Tekrar deneniyor';

SELECT * FROM DatabaseVersion ORDER BY VersionID DESC;

```

Completion time: 2025-05-30T08:10:25.8344109+03:00

Messages

Msg 2628, Level 16, State 1, Procedure pr\_ApplyVersion, Line 46 [Batch Start Line 24]  
String or binary data would be truncated in table 'ornekdb.dbo.DatabaseVersion', column 'Status'. Truncated value: 'Failed - Bu versiyon zaten uygulanmış veya daha ye'.  
The statement has been terminated.  
Msg 50000, Level 16, State 1, Procedure pr\_ApplyVersion, Line 50 [Batch Start Line 24]  
Versiyon uygulama başarısız oldu.

Şekil 46

Şekil 46'da sonradan eklenen versiyon bir daha çalıştırıldığında hata alındığı gözlenmektedir.

110 %

Results Messages

	VersionID	VersionNumber	Description	Status	AppliedDate
1	6	1.2.1	Customers1 tablosuna Phone sütunu eklendi.	Completed	2025-05-30 08:10:17.863
2	5	1.2.0	Customers1 tablosu oluşturuldu.	Completed	2025-05-30 08:06:46.260
3	2	1.1.0	Created Customers table	Completed	2025-05-30 06:31:00.977
4	1	1.0.0	Initial setup - version system created	Completed	2025-05-30 06:29:20.070

Şekil 47

Şekil 47'de yapılan değişiklikler databaseversion tablosunda görülmektedir.

The screenshot shows a SQL query window in SSMS. The query is:

```
EXEC pr_RollbackVersion '1.2.0';
SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'Customers3';
```

The results pane displays the columns of the 'Customers3' table:

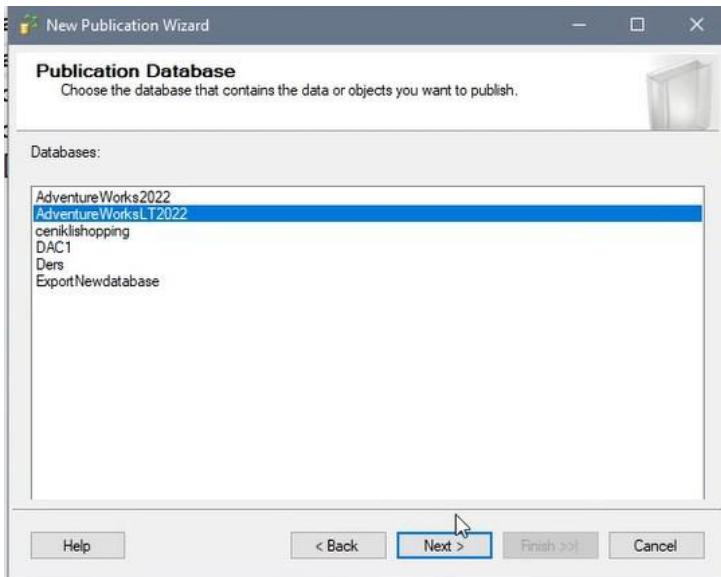
	COLUMN_NAME
1	CustomerID
2	FirstName
3	LastName
4	Email

Şekil 48

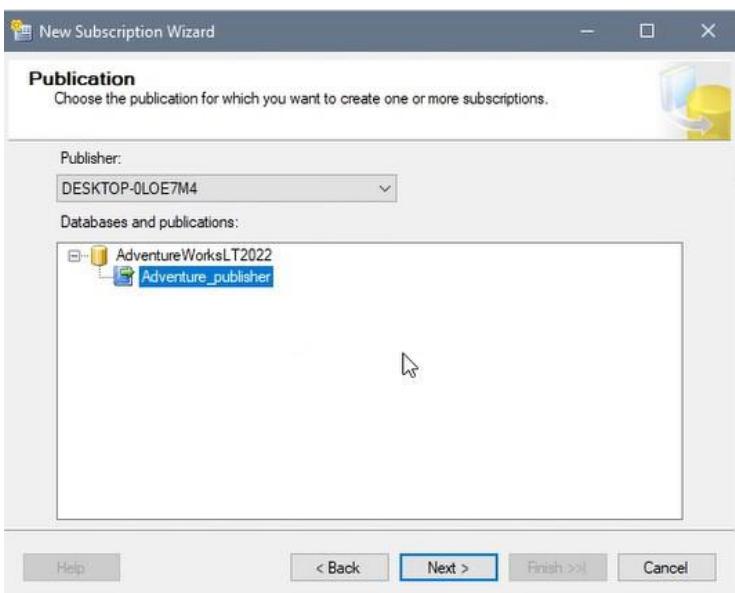
Şekil 48'de bir önceki versiyona rollback yapılmınca sonradan customers3 tablosuna eklediğimiz phone sütünunu kalktığı görülmektedir.

## 7. Veritabanı Yük Dengeleme ve Dağıtık Veritabanı Yapıları

## a. Veritabanı Replikasyonu



Listeden AdventureWorksLT2022 veritabanı seçilmiştir. Bu veritabanı, replikasyon için veri kaynağı olacaktır.

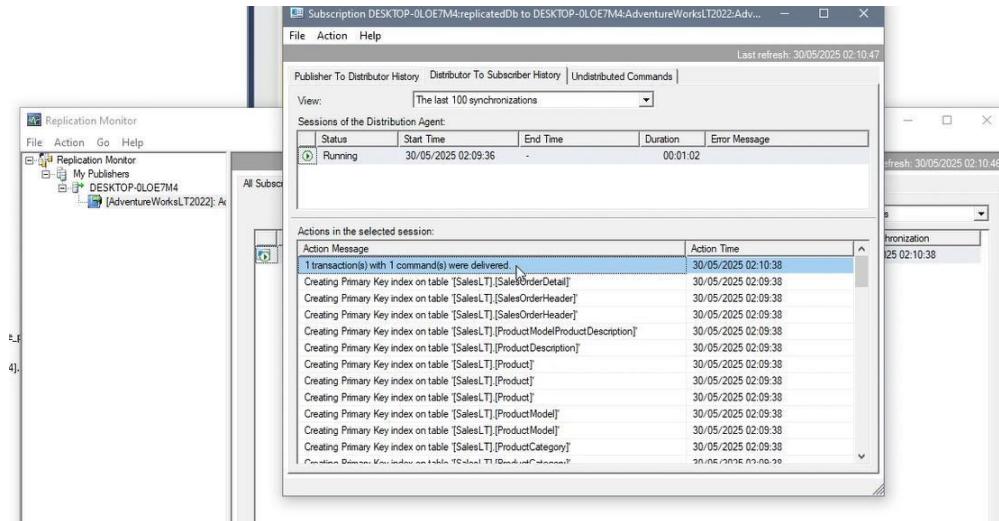


Daha önce oluşturulan AdventureWorksLT2022 -> Adventure\_publisher yayını seçilmiştir. Bu, abonenin belirli bir yayıcısından ve yayından veri tüketeceğini doğrular.



Bu resim "Replikasyon İzleyici"yi (Replication Monitor) göstermektedir.

[DESKTOP-0LOE7M4]: replicatedDb aboneliğinin Çalışıyor (Running) durumda olduğunu, "Mükemmel" (Excellent) performansla ve 00:00:06 (6 saniye) gecikmeyle çalıştığını belirtir.



[SalesLT].[ProductModel] tablosundaki bir satırı değiştirmek için bir UPDATE deyimi yürütülmektedir. Özellikle, ProductModelID = 1 olan kaydın Name değeri 'New Mountain Bike Elite', CatalogDescription boş, ve rowguid ile ModifiedDate alanları güncellenmektedir. Bu, yayınıcı üzerinde aboneye çoğaltıması gereken bir değişikliği simüle eder.

The screenshot shows an SSMS query editor window with the title 'SQLQuery1.sql - D...0LOE7M4\emre (69)\*'. The code is as follows:

```
USE [AdventureWorksLT2022]
GO

-- Belirli bir ProductModelID'yi güncelleme
UPDATE [SalesLT].[ProductModel]
SET
    [Name] = 'New Mountain Bike Elite',
    [CatalogDescription] = '',
    [rowguid] = NEWID(),
    [ModifiedDate] = GETDATE()
WHERE [ProductModelID] = 1 -- Güncellenecek kaydın ID'si
GO
```

Bu resim tekrar "Replikasyon İzleyici"yi, özellikle "Yayından Dağıtıcıya Geçmiş" (Publisher To Distributor History) sekmesini göstermektedir.

Dağıtım Aracısı'nın (Distribution Agent) "Çalışıyor" (Running) bir oturumunu göstermektedir.

En önemlisi, "Eylem Mesajı" (Action Message) altında "1 transaction(s) with 1 command(s) were delivered" (1 işlem, 1 komut ile teslim edildi) ifadesi yer almaktadır.