

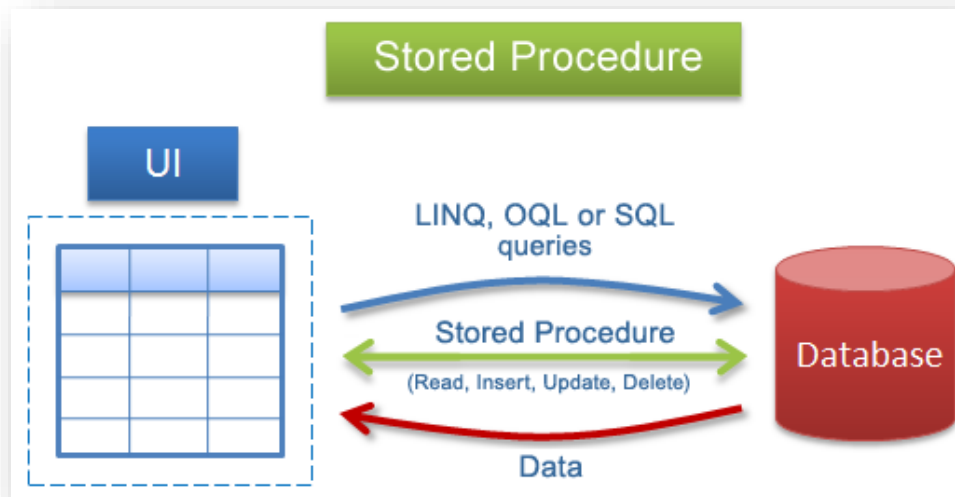


VERİTABANI YÖNETİM SİSTEMLERİ

Dr. Öğr. Üyesi
Ender Şahinaslan

BÖLÜM -10-

STORED PROCEDURE



GENEL BAKIŞ...

Stored Procedure (SP)

- Extended Stored Procedure
- CLR Stored Procedure'ler
- Sistem Stored Procedure'leri
- Kullanıcı Tanımlı Stored Procedure'ler

Stored Procedure'lerin Çalıştırılması

Stored Procedure'lerin Oluşturulması

Stored Procedure'lerde Parametre Kullanmak

10.1. STORED PROCEDURE

Prosedür, belli bir görevi yerine getirmek için yazılmış program parçacıklarıdır. Başka bir deyişle, herhangi bir işlevi yerine getirmek için yazılan kodların bir paket içerisinde tutulmuş halidir.

Bir prosedür başka bir prosedür tarafından da çağrılabilir. Bu, sık kullanılan işlemlerin bir defa yazılarak programın akışına göre defalarca kullanılmasına olanak sağlamaktadır. Böylelikle kod yazımı ve programlama işlemi kolaylaştırılmış olmaktadır.

10.1. STORED PROCEDURE (DEVAM...)

Stored Procedure'ler diğer programlama dillerindeki **fonksiyonlarla aynı işleve** sahiptir.

Oluşturulan bir **Stored Procedure'e** ana **program içerisinde bir komut ile ulaşılabilir.**

Aynı komut defalarca verilerek procedure çalıştırılabilir. **Dolayısı ile procedure defalarca yazılmayıp tek bir defa yazılıp çağırılmış olmaktadır.**

10.1. STORED PROCEDURE (DEVAM...)

Stored Procedure'ler, **ilk defa 1980'li yıllarda Sybase veritabanı sunucusunda** kullanılmıştır.

Sorguların önceden hazırlanması ve veritabanı yönetim sistemi içerisinde saklanması en büyük özelliğidir. **Dolayısı ile aynı uzayda çalıştığı için daha hızlı sonuçlar** alınabilmektedir.

Bir **SP oluşturulduktan sonra veritabanı içerisinde saklanmaktadır** ve **ihtiyaç duyulduğunda ise çağrılmaktadır.**

10.1. STORED PROCEDURE (DEVAM...)

Stored Procedure'ler, **oluşturma şekillerine göre dört'e ayrılır:**

- **Extended** Stored Procedure'ler
- **CLR** Stored Procedure'ler
- **Sistem** Stored Procedure'leri
- **Kullanıcı Tanımlı** Stored Procedure'ler

10.1.1. EXTENDED STORED PROCEDURE

Genellikle **.dll** uzantılı derlenmiş olan prosedürlerdir.

T-SQL dışındaki C, C++, #C, Basic, Delphi gibi dillerde yazılıp derlenirler.

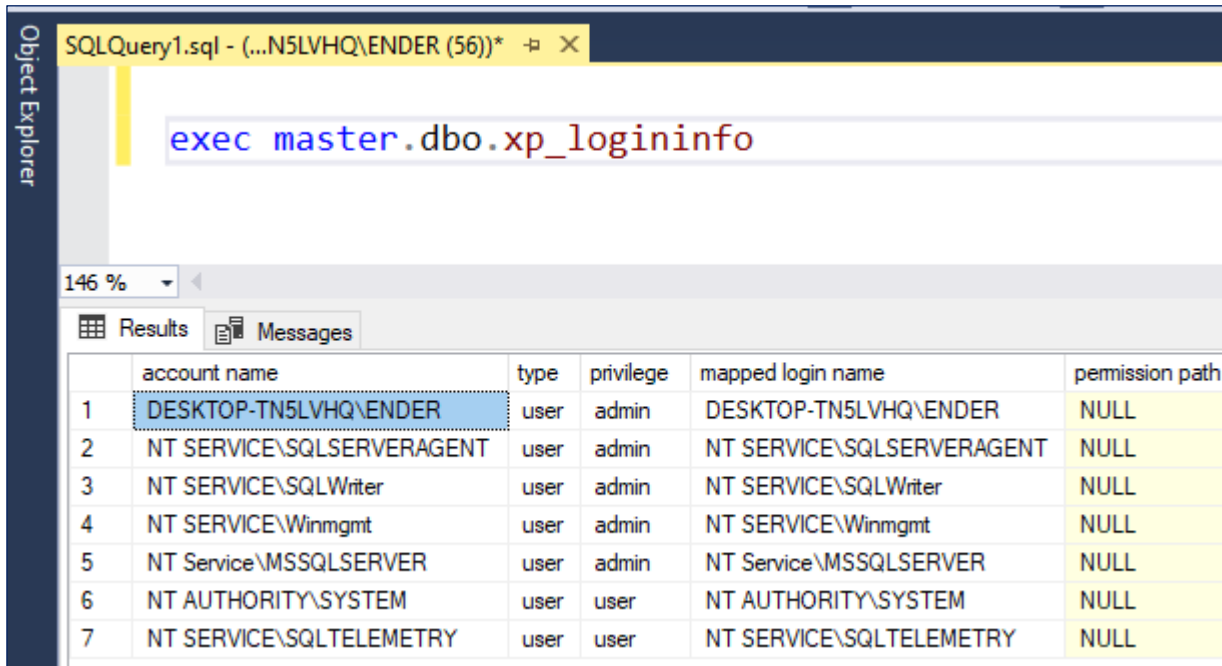
Veritabanı uzayı dışında çalışırlar.

Sistem içerisinde hazır olarak gelen **extended procedure**'ler genellikle **xp_ öneki** ile başlarlar, fakat **sp_ öneki** ile başlayan prosedürlerde vardır.

Bu sistem **extended prosedürleri** **master veritabanı içerisinde** tutulmaktadır. Tam nitelikli isimleri **db.sahibi.sp-ismi** şeklindedir.

10.1.1. EXTENDED STORED PROCEDURE (DEVAM...)

Bütün sistem prosedürleri master veritabanı altında görülebilir. Örneğin veritabanına bağlanan kullanıcıları listelemek için aşağıdaki extended sistem prosedürü kullanılabilir.



SQLQuery1.sql - (...N5LVHQ\ENDER (56))*

```
exec master.dbo.xp_logininfo
```

146 %

Results Messages

	account name	type	privilege	mapped login name	permission path
1	DESKTOP-TN5LVHQ\ENDER	user	admin	DESKTOP-TN5LVHQ\ENDER	NULL
2	NT SERVICE\SQLSERVERAGENT	user	admin	NT SERVICE\SQLSERVERAGENT	NULL
3	NT SERVICE\SQLWriter	user	admin	NT SERVICE\SQLWriter	NULL
4	NT SERVICE\Winmgmt	user	admin	NT SERVICE\Winmgmt	NULL
5	NT Service\MSSQLSERVER	user	admin	NT Service\MSSQLSERVER	NULL
6	NT AUTHORITY\SYSTEM	user	user	NT AUTHORITY\SYSTEM	NULL
7	NT SERVICE\SQLTELEMETRY	user	user	NT SERVICE\SQLTELEMETRY	NULL

10.1.2. CLR STORED PROCEDURE

CLR (Common Language Runtime), SQL Server 2005 sonrasında CLR ortamında herhangi bir dili kullanarak kodlanan SP'lerdir.

CLR ve .NET ile SP'ler, kullanıcı tanımlı fonksiyonlar, ve Trigger'lar gibi nesneleri kodlamak mümkündür. Dolayısı ile SP ve diğer nesneleri kodlamak için CLR'ye ihtiyaç vardır. Bununla birlikte SP için çeşitli CLR prosedürleri'de beraberinde gelmektedir.

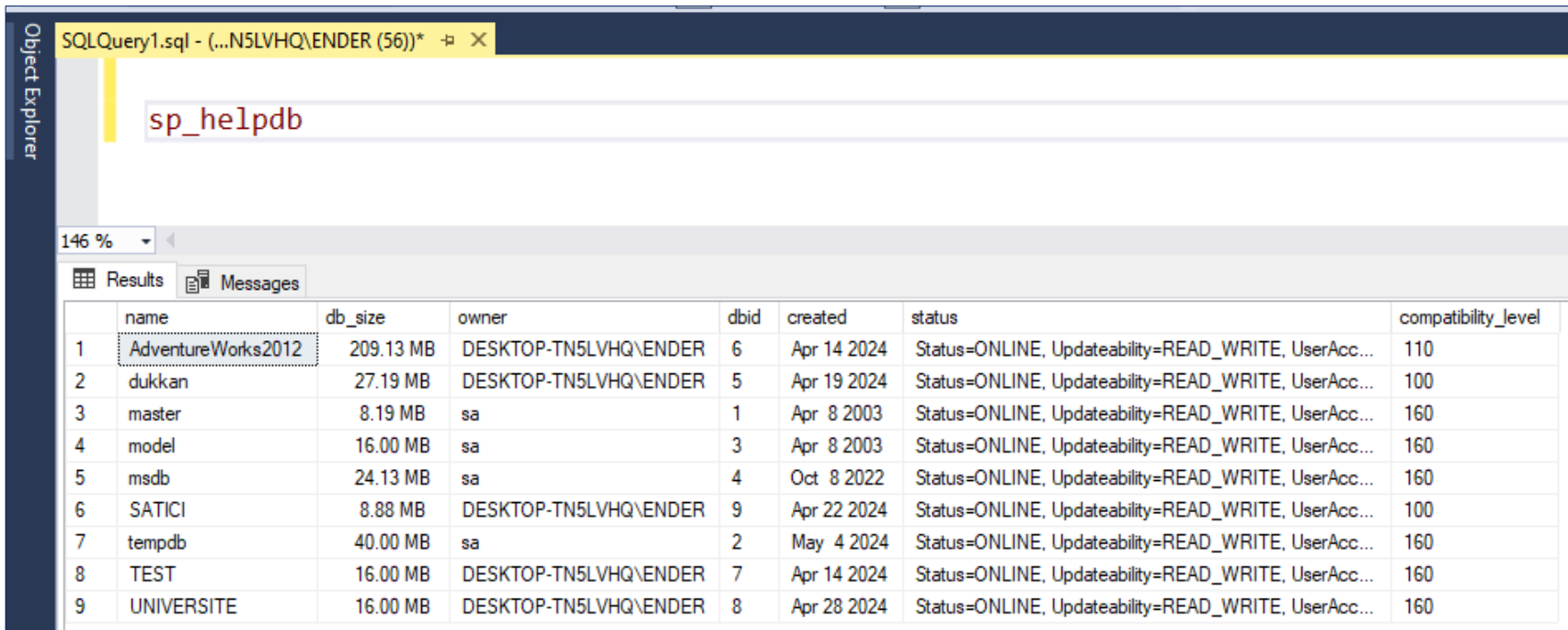
Güvenlik ve denetim gibi konularda CLR SP'ler, SQL Server için büyük avantaj sağlamaktadır.

10.1.3. SİSTEM STORED PROCEDURE

- Genellikle **sp_** ön ekiyle başlar.
- **Master** veri tabanında tutulurlar.
- Extended Stored Procedure'de olduğu gibi tam adlarının yazılması zorunluluğu yoktur. Sadece adını yazmak yeterlidir.
- İstenirse tam ismi kullanılarak ta çalıştırılabilir fakat böyle bir zorunluluk yoktur.

10.1.3. SİSTEM STORED PROCEDURE (DEVAM...)

Örneğin **sp_helpdb** sistem prosedürü veritabanı yönetim sisteminde yer alan veritabanlarını listeler.



Object Explorer

SQLQuery1.sql - (...N5LVHQ\ENDER (56))*

sp_helpdb

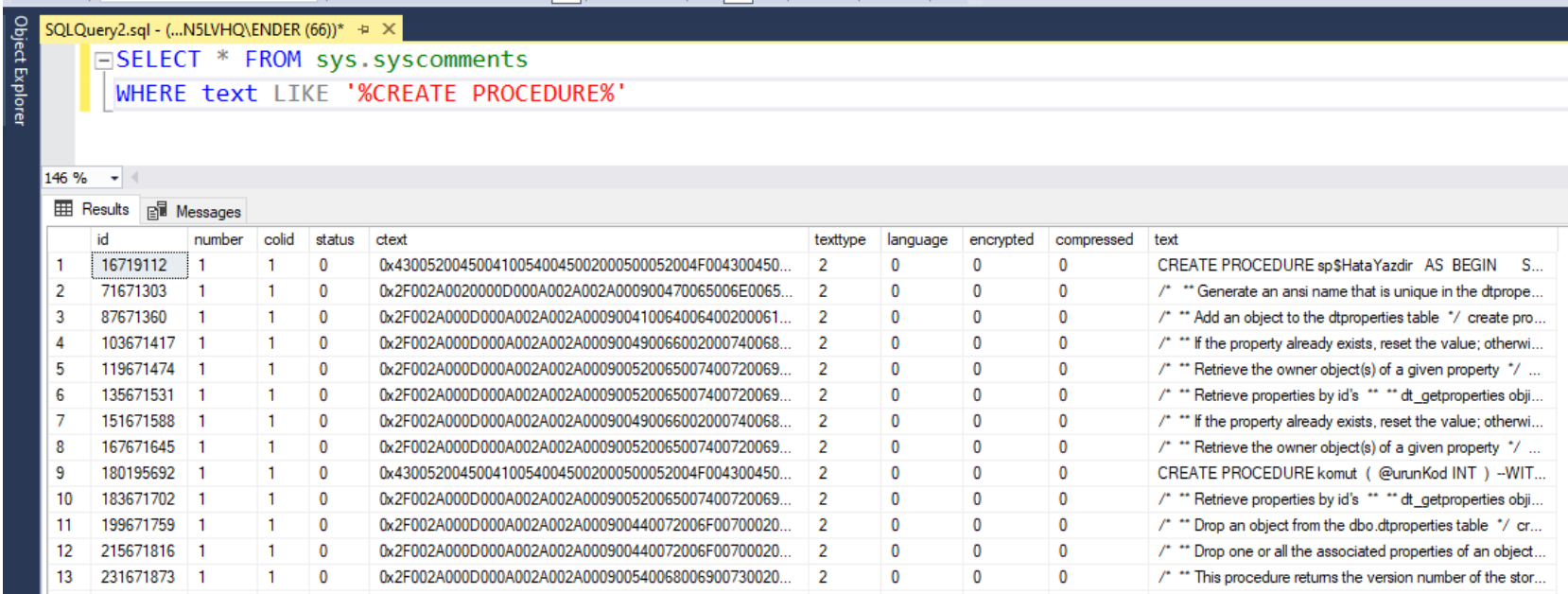
146 %

Results Messages

	name	db_size	owner	dbid	created	status	compatibility_level
1	AdventureWorks2012	209.13 MB	DESKTOP-TN5LVHQ\ENDER	6	Apr 14 2024	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	110
2	dukkani	27.19 MB	DESKTOP-TN5LVHQ\ENDER	5	Apr 19 2024	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	100
3	master	8.19 MB	sa	1	Apr 8 2003	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	160
4	model	16.00 MB	sa	3	Apr 8 2003	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	160
5	msdb	24.13 MB	sa	4	Oct 8 2022	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	160
6	SATICI	8.88 MB	DESKTOP-TN5LVHQ\ENDER	9	Apr 22 2024	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	100
7	tempdb	40.00 MB	sa	2	May 4 2024	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	160
8	TEST	16.00 MB	DESKTOP-TN5LVHQ\ENDER	7	Apr 14 2024	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	160
9	UNIVERSITE	16.00 MB	DESKTOP-TN5LVHQ\ENDER	8	Apr 28 2024	Status=ONLINE, Updateability=READ_WRITE, UserAcc...	160

10.1.3. SİSTEM STORED PROCEDURE (DEVAM...)

SQL Server üzerindeki nesnelerin tanımları, sys.syscomments catalog view'inde durur. Doğrudan bu view'i sorgulamak istediğimiz nesneye daha hızlı ulaşmasını sağlar. Örneğin aşağıdaki ifade o database'de bulunan stored procedure'lerin tanımlarını gösterir.



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query in SQLQuery2.sql: `SELECT * FROM sys.syscomments WHERE text LIKE '%CREATE PROCEDURE%'`. The bottom pane shows the results of this query, which is a table with columns: id, number, colid, status, ctext, texttype, language, encrypted, compressed, and text. The results list 13 rows of stored procedure definitions from the sys.syscomments catalog view.

	id	number	colid	status	ctext	texttype	language	encrypted	compressed	text
1	16719112	1	1	0	0x4300520045004100540045002000500052004F004300450...	2	0	0	0	CREATE PROCEDURE sp\$HataYazdir AS BEGIN S...
2	71671303	1	1	0	0x2F002A002000D000A002A002A000900470065006E0065...	2	0	0	0	/* ** Generate an ansi name that is unique in the dtprope...
3	87671360	1	1	0	0x2F002A000D000A002A002A000900410064006400200061...	2	0	0	0	/* ** Add an object to the dtproperties table */ create pro...
4	103671417	1	1	0	0x2F002A000D000A002A002A000900490066002000740068...	2	0	0	0	/* ** If the property already exists, reset the value; otherwi...
5	119671474	1	1	0	0x2F002A000D000A002A002A000900520065007400720069...	2	0	0	0	/* ** Retrieve the owner object(s) of a given property */ ...
6	135671531	1	1	0	0x2F002A000D000A002A002A000900520065007400720069...	2	0	0	0	/* ** Retrieve properties by id's ** dt_getproperties obji...
7	151671588	1	1	0	0x2F002A000D000A002A002A000900490066002000740068...	2	0	0	0	/* ** If the property already exists, reset the value; otherwi...
8	167671645	1	1	0	0x2F002A000D000A002A002A000900520065007400720069...	2	0	0	0	/* ** Retrieve the owner object(s) of a given property */ ...
9	180195692	1	1	0	0x4300520045004100540045002000500052004F004300450...	2	0	0	0	CREATE PROCEDURE komut (@urunKod INT) -WIT...
10	183671702	1	1	0	0x2F002A000D000A002A002A000900520065007400720069...	2	0	0	0	/* ** Retrieve properties by id's ** dt_getproperties obji...
11	199671759	1	1	0	0x2F002A000D000A002A002A000900440072006F00700020...	2	0	0	0	/* ** Drop an object from the dbo.dtproperties table */ cr...
12	215671816	1	1	0	0x2F002A000D000A002A002A000900440072006F00700020...	2	0	0	0	/* ** Drop one or all the associated properties of an object...
13	231671873	1	1	0	0x2F002A000D000A002A002A000900540068006900730020...	2	0	0	0	/* ** This procedure returns the version number of the stor...

10.1.4. KULLANICI TANIMLI STORED PROCEDURE

- Programcı tarafından programlanan prosedürlerdir.
- Konu olarak en fazla üzerinde duracağımız prosedürlerdir.
- Geçerli oldukları duruma göre üç'e ayrılırlar:
 - Geçici SP'ler,
 - Yerel SP'ler,
 - Uzak SP'ler.

10.1.4. KULLANICI TANIMLI STORED PROCEDURE (DEVAM...)

Geçici Stored Procedure'ler, özellikle SQL Server'ın eski sürümlerinde kullanılan bir türdür.

- Her oturum açıldığında derlenmesi gereken SP'lerdir.

Yerel SP'ler ise, kullanıcı tarafından oluşturulan ve bu ders boyunca üzerinde duracağımız SP'lerdir.

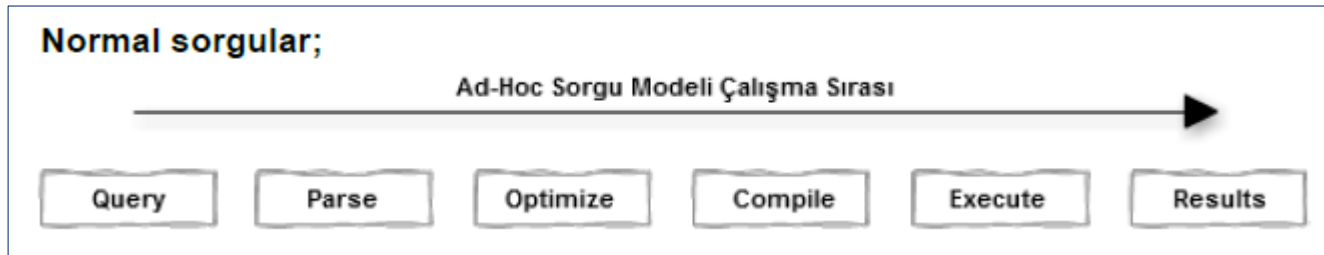
Uzak SP'ler, dağıtık bir modelde tasarlanan ve uzaktaki bir sunucuda yer alan SP'lerdir.

10.2. SP'LERİN ÇALIŞTIRILMASI

SP'lerin performans açısından ne sağladığını anlamak için kodlama aşamasından çalıştırılma aşamasına kadar bütün aşamaları çok iyi anlamak ve bilmek gerekmektedir.

Bu aşamaları şöyle sıralayabiliriz:

- Ayrıştırma (Parsing),
- Derleme (Compiling),
- Çalıştırma (Executing).



10.2. SP'LERİN ÇALIŞTIRILMASI (DEVAM...)

- Ayrıştırma işlemi sırasında **SQL ifadelerinin geçerli olup olmadığı** denetlenir.
- Veritabanı içinde tablo, VIEW gibi başka nesneler varsa ve bunlara gönderme yapılıyorsa, bu **nesnelere ait izinlerin olup olmadığı** kontrol edilir.
- Sorgu ağacı yada sıra ağacı denilen bir yapı ortaya çıkarılır.
- Ardından **SP'nin ismi** **sysobjects** tablosuna kaydedilir. Daha sonra da **syscomments** tablosuna **SP'nin kodları** kaydedilir.

10.2. SP'LERİN ÇALIŞTIRILMASI (DEVAM...)

- **Derleme aşamasında,** bir önceki aşamada oluşturulan sıra ağacı ele alınarak bir çalışma planı çıkartılır.
- Bütün çalışma planı üzerinde **güvenlik ve yetkiler** denetlenir.
- **Bu çalışma planı,** hangi aşamada hangi kontrollerin kullanılacağını veya tabloların kontrol edileceğini içermektedir.

10.2. SP'LERİN ÇALIŞTIRILMASI (DEVAM...)

Çalıştırma aşamasında, bir önceki aşamada elde edilen çalışma planına göre çalıştırma işlemi gerçekleştirilir.

- Örneğin bir SELECT ifadesi kullanılmışsa sorgu veri işlemeden sorumlu DML (Data Manipulation Language) yöneticisine iletilir.

Bir SP eğer ilk defa çağrılıyorsa, bu **üç aşama gerçekleştirilir**. Eğer daha önce çalıştırıldıysa sorgulama ağacı oluşturma işlemleri yapılmaz ve oldukça **hızlı bir şekilde SP'nin derlenmiş hali çalıştırılır**. Bunun nedeni SP'lerin derlenmiş birer nesne olmasıdır.

10.3. SP'LERİN OLUŞTURULMASI

Bir kullanıcı tanımlı prosedür oluşturmanın genel biçimi aşağıda verilmektedir.

```
CREATE PROC/PROCEDURE prosedur_adi
```

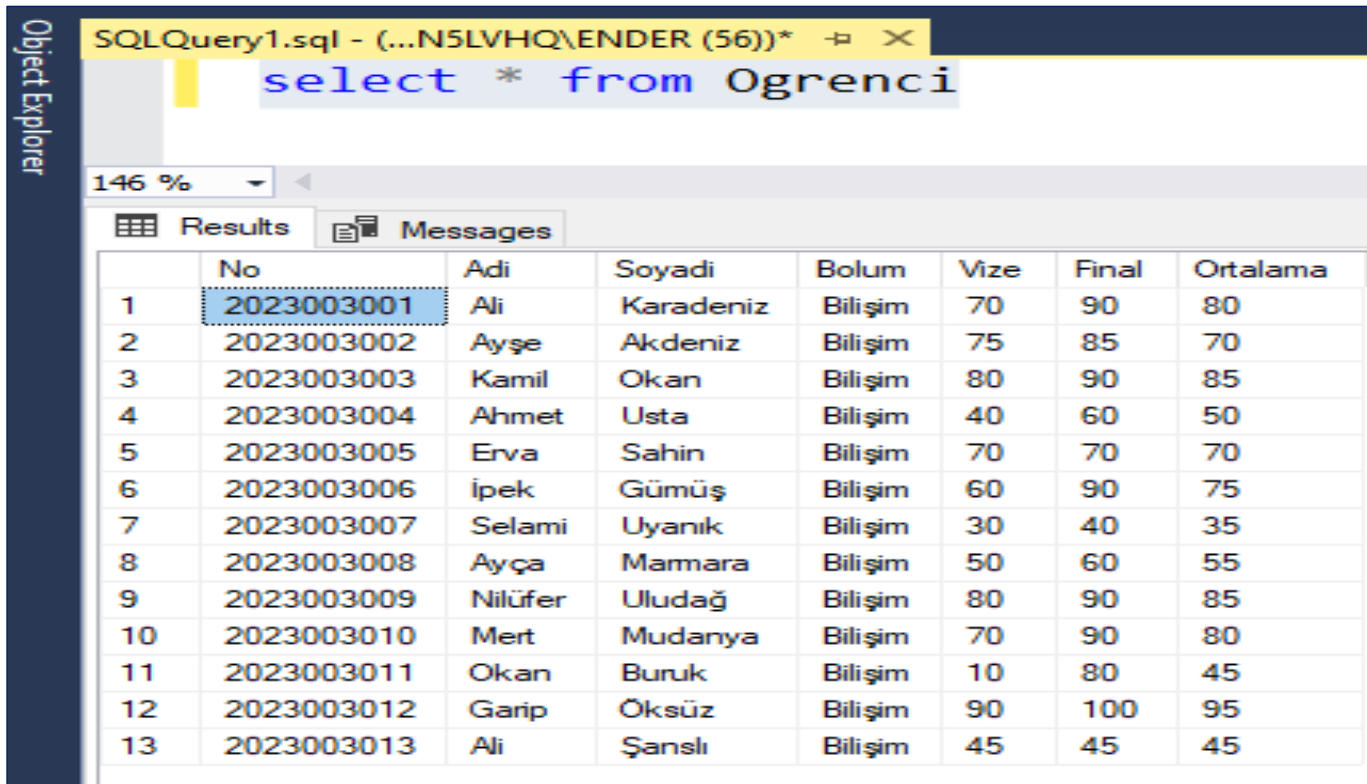
```
AS
```

```
SQL ifadeleri
```

NOT: SP oluştururken PROC deyimi yada
PROCEDURE deyimi kullanılmaktadır.

10.3. SP'LERİN OLUŞTURULMASI (DEVAM...)

Örnek-1: Öğrenciler tablosu aşağıda verilmektedir. Bu tabloyu göz önüne alarak ortalaması 70'in üzerinde olan öğrencileri bulan bir Stored Procedure yazalım.

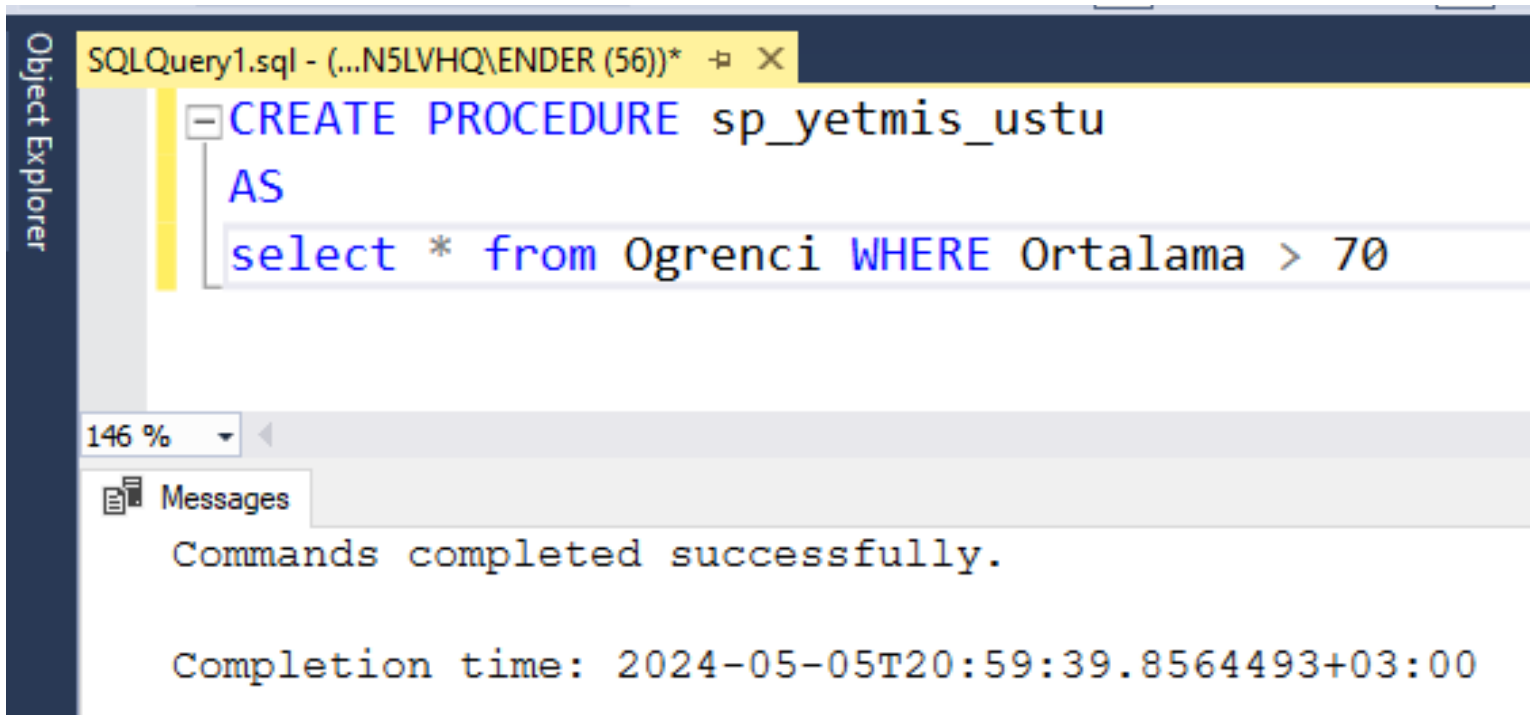


The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays 'Object Explorer'. The main pane shows a query window titled 'SQLQuery1.sql - (...N5LVHQ\ENDER (56))*'. The query text is 'select * from Ogrenci'. Below the query window, the 'Results' tab is active, displaying a table with 13 rows and 8 columns. The columns are 'No', 'Adi', 'Soyadi', 'Bolum', 'Vize', 'Final', and 'Ortalama'. The first row is highlighted with a blue border.

	No	Adi	Soyadi	Bolum	Vize	Final	Ortalama
1	2023003001	Ali	Karadeniz	Bilişim	70	90	80
2	2023003002	Ayşe	Akdeniz	Bilişim	75	85	70
3	2023003003	Kamil	Okan	Bilişim	80	90	85
4	2023003004	Ahmet	Usta	Bilişim	40	60	50
5	2023003005	Erva	Sahin	Bilişim	70	70	70
6	2023003006	İpek	Gümüş	Bilişim	60	90	75
7	2023003007	Selami	Uyanık	Bilişim	30	40	35
8	2023003008	Ayça	Mamara	Bilişim	50	60	55
9	2023003009	Nilüfer	Uludağ	Bilişim	80	90	85
10	2023003010	Mert	Mudanya	Bilişim	70	90	80
11	2023003011	Okan	Buruk	Bilişim	10	80	45
12	2023003012	Garip	Öksüz	Bilişim	90	100	95
13	2023003013	Ali	Şanslı	Bilişim	45	45	45

10.3. SP'LERİN OLUŞTURULMASI (DEVAM...)

Yazılacak SP kodu aşağıdaki gibidir:



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane is visible. The main window shows a query editor with the following SQL code:

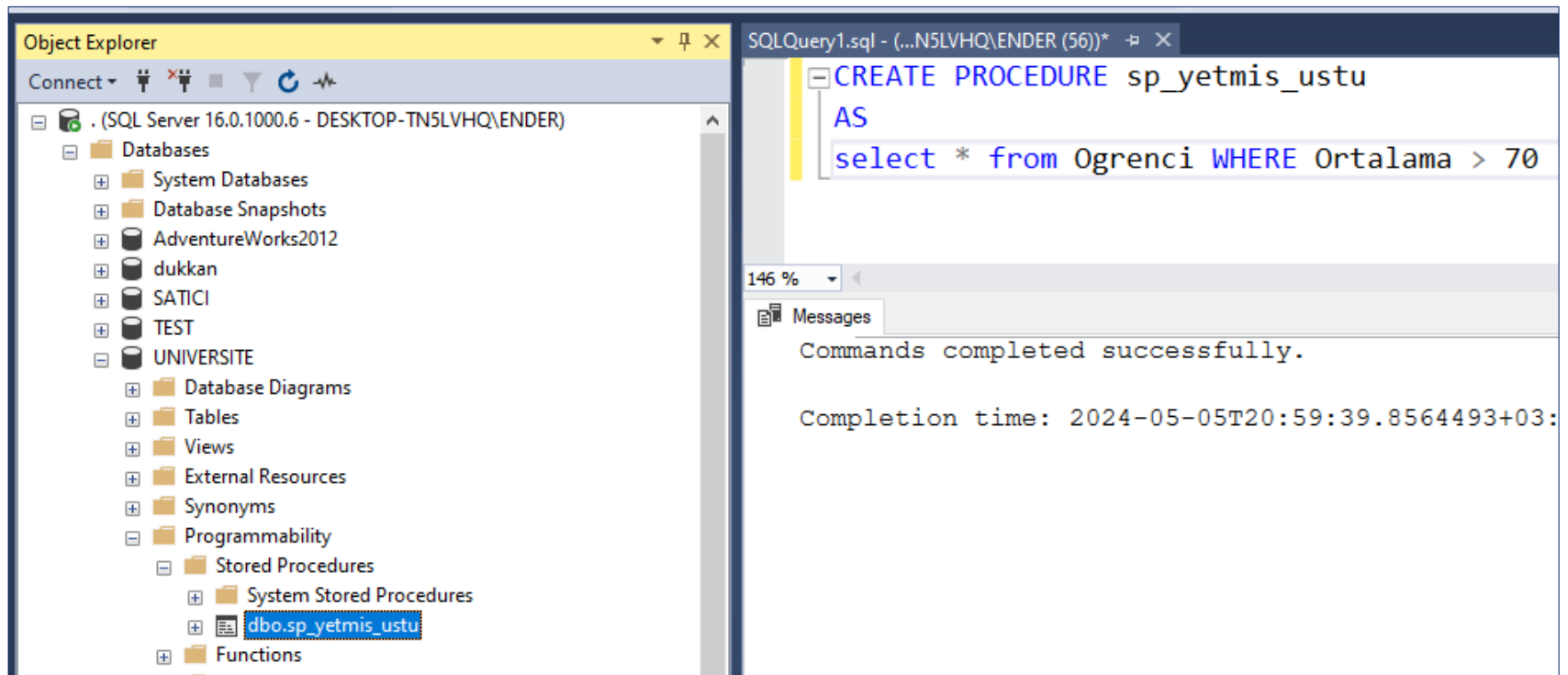
```
SQLQuery1.sql - (...N5LVHQ\ENDER (56))*  
CREATE PROCEDURE sp_yetmis_ustu  
AS  
select * from Ogrenci WHERE Ortalama > 70
```

Below the query editor, the 'Messages' pane shows the execution results:

```
Commands completed successfully.  
  
Completion time: 2024-05-05T20:59:39.8564493+03:00
```

10.3. SP'LERİN OLUŞTURULMASI (DEVAM...)

Oluşturulan SP (sp_yetmiş_ustu) obje ve kodu:



10.3. SP'LERİN OLUŞTURULMASI (DEVAM...)

Oluşturulan bir SP'nin çalıştırılmasında iki yol vardır. Ya **Doğrudan SP'nin ismi** yada **exec SP_adi** yazarak çalıştırabilirsiniz. Örneğin:

Object Explorer

SQLQuery4.sql - (...N5LVHQ\ENDER (60))* X SQLQuery1.sql - (...N5LVHQ\ENDER (60))*

sp_yetmis_ustu

146 %

Results Messages

	No	Adi	Soyadi	Bolum	Vize	Final	Ortalama
1	2023003001	Ali	Karadeniz	Bilişim	70	90	80
2	2023003003	Kamil	Okan	Bilişim	80	90	85
3	2023003006	İpek	Gümüş	Bilişim	60	90	75
4	2023003009	Nilüfer	Uludağ	Bilişim	80	90	85
5	2023003010	Mert	Mudanya	Bilişim	70	90	80
6	2023003012	Garip	Öksüz	Bilişim	90	100	95

Object Explorer

SQLQuery4.sql - (...N5LVHQ\ENDER (60))* X SQLQuery1.sql - (...N5LVHQ\ENDER (60))*

exec sp_yetmis_ustu

146 %

Results Messages

	No	Adi	Soyadi	Bolum	Vize	Final	Ortalama
1	2023003001	Ali	Karadeniz	Bilişim	70	90	80
2	2023003003	Kamil	Okan	Bilişim	80	90	85
3	2023003006	İpek	Gümüş	Bilişim	60	90	75
4	2023003009	Nilüfer	Uludağ	Bilişim	80	90	85
5	2023003010	Mert	Mudanya	Bilişim	70	90	80
6	2023003012	Garip	Öksüz	Bilişim	90	100	95

10.3. SP'LERİN OLUŞTURULMASI (DEVAM...)

Örnek-2: Gün içerisinde yapılan satışları bulan bir SP oluşturalım. Bunun için **dukkana VT** üzerinde yer alan **tblSiparis** ve **tblSiparisDetay** tablo detayları aşağıda verilmiştir.

tblSiparis

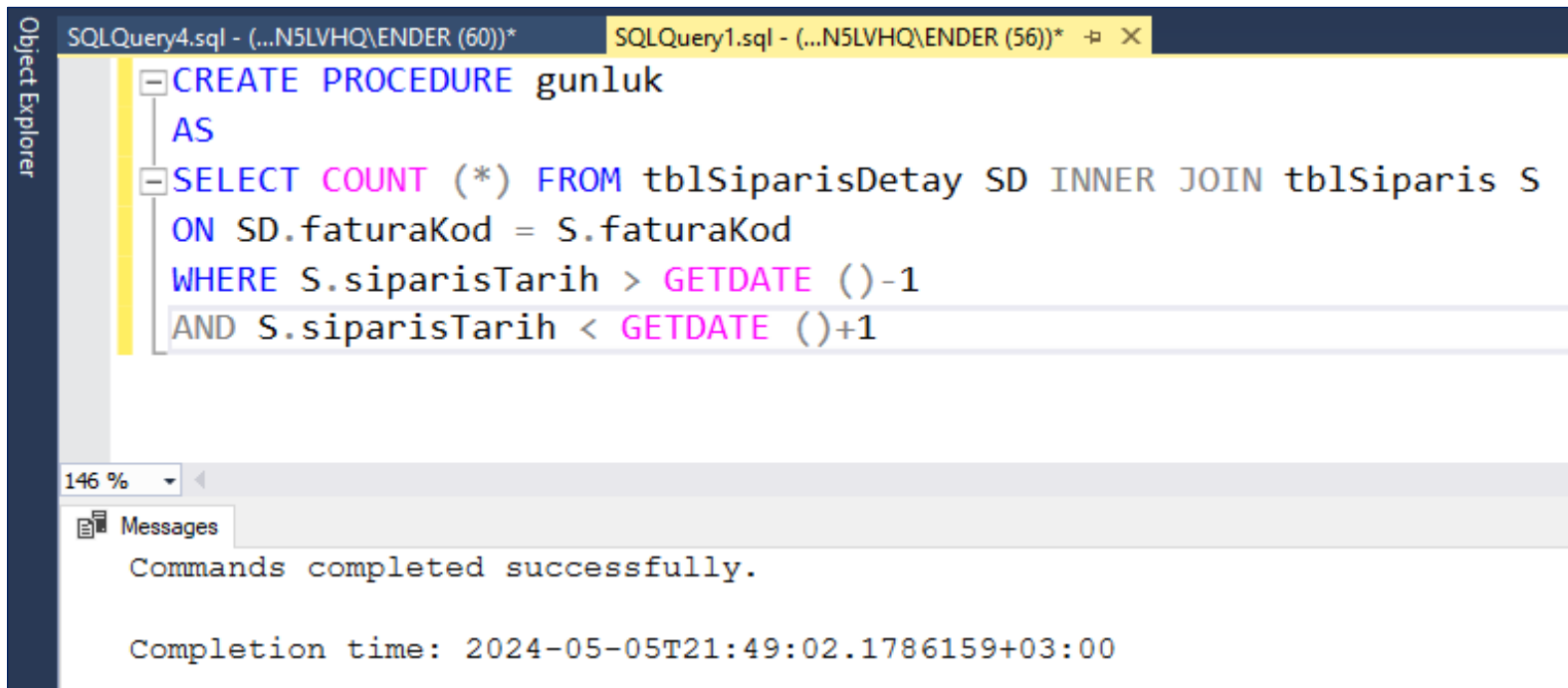
	Column Name	Data Type	Allow Nulls
▶	FaturaKod	int	<input type="checkbox"/>
	kullaniciKod	int	<input checked="" type="checkbox"/>
	faturaAdresi	nvarchar(255)	<input checked="" type="checkbox"/>
	vergiDairesi	nvarchar(50)	<input checked="" type="checkbox"/>
	vergiNo	nvarchar(50)	<input checked="" type="checkbox"/>
	siparisDurumKod	int	<input checked="" type="checkbox"/>
	siparisTarih	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

tblSiparisDetay

	Column Name	Data Type	Allow Nulls
▶	SatirNo	int	<input type="checkbox"/>
	faturaKod	int	<input checked="" type="checkbox"/>
	urunKod	int	<input checked="" type="checkbox"/>
	fiyat	float	<input checked="" type="checkbox"/>
	KDV	float	<input checked="" type="checkbox"/>
	adet	smallint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

10.3. SP'LERİN OLUŞTURULMASI (DEVAM...)

SP aşağıdaki gibi oluşturulabilir.



The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane is visible. The main window shows a SQL query editor with the following code:

```
CREATE PROCEDURE gunluk
AS
SELECT COUNT (*) FROM tblSiparisDetay SD INNER JOIN tblSiparis S
ON SD.faturaKod = S.faturaKod
WHERE S.siparisTarih > GETDATE ()-1
AND S.siparisTarih < GETDATE ()+1
```

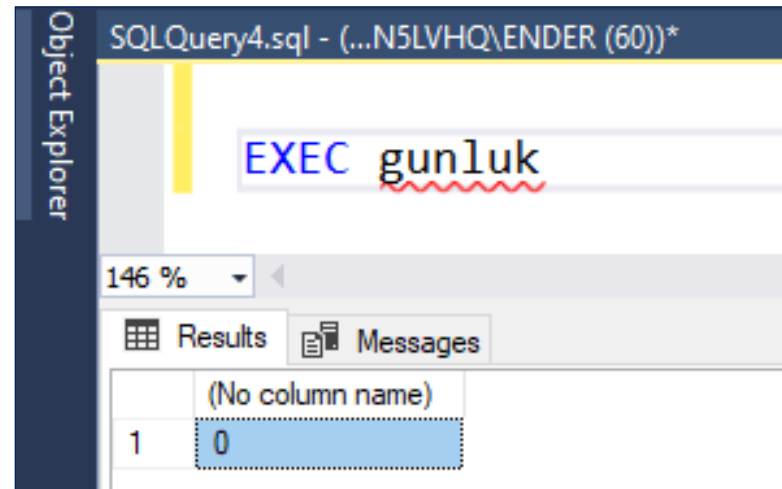
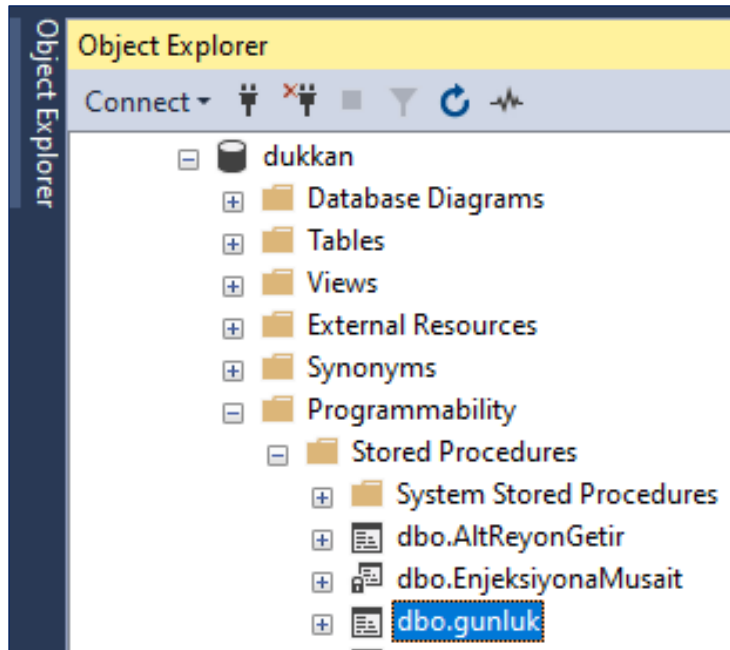
Below the query editor, the 'Messages' pane shows the following output:

```
Commands completed successfully.

Completion time: 2024-05-05T21:49:02.1786159+03:00
```

10.3. SP'LERİN OLUŞTURULMASI (DEVAM...)

Oluşturulan 'gunluk' SP çalıştırıldığında günlük satış yapıp yapılmadığı aşağıda görülmektedir.



10.4. SP'LERDE DEĞİŞİKLİK YAPMAK

- Bir SP'de değişiklik yapmak için **ALTER** komutu kullanılmaktadır.
- ALTER komutu ile geçen dersimizde anlatmış olduğumuz **WITH ENCRYPTION** komutu da kullanılabilir. Bildiğiniz gibi **WITH ENCRYPTION** komutu kodların başkaları tarafından görülmesini engellemektedir.
- **ALTER** komutunun genel kullanım biçimi aşağıdaki gibidir:

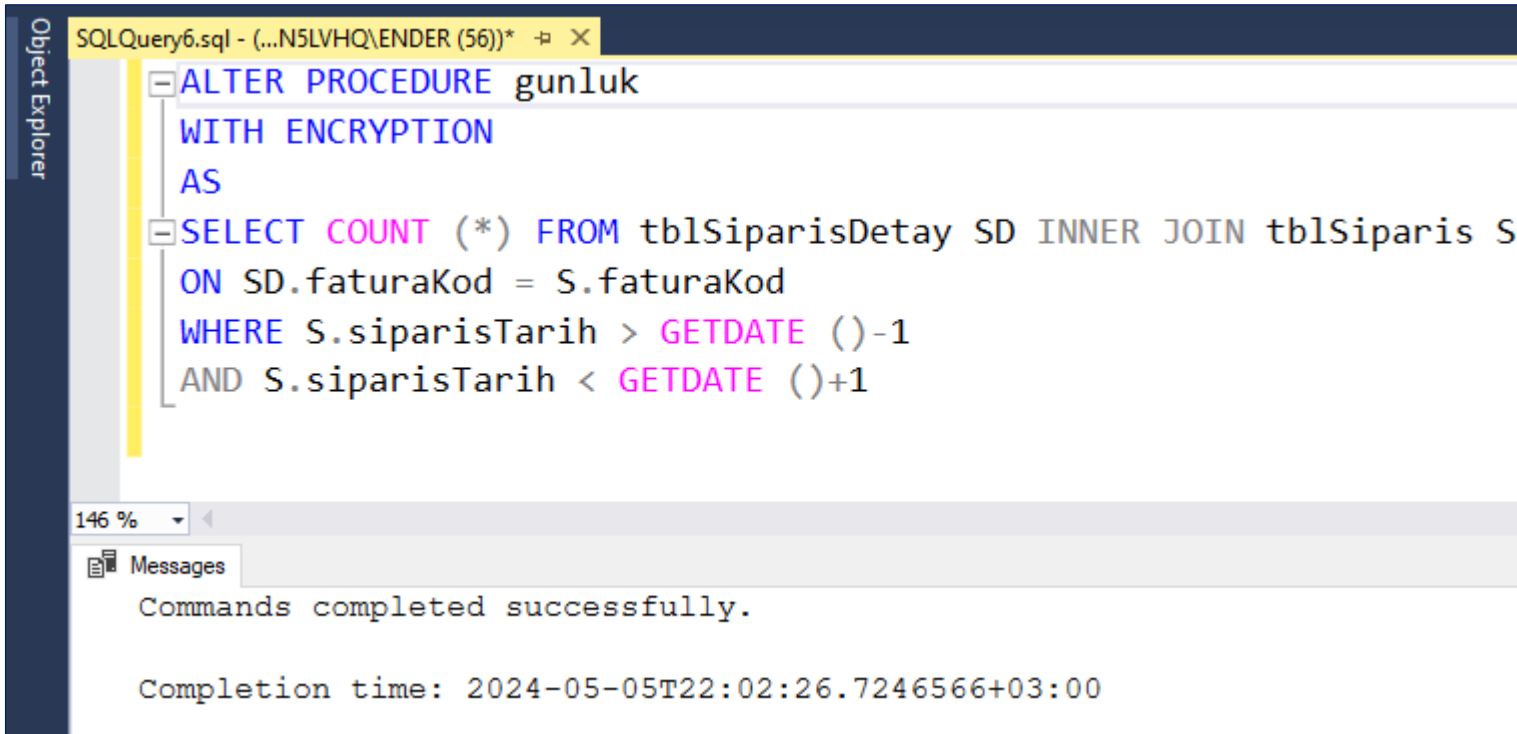
ALTER PROCEDURE Prosedür_ismi

AS

SQL ifadeleri

10.4. SP'LERDE DEĞİŞİKLİK YAPMAK (DEVAM...)

Daha önce oluşturduğumuz günlük satış SP'si üzerinde şifreleyerek değişiklik yapalım.



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL code for altering the 'gunluk' stored procedure with encryption. The bottom pane shows the execution results, indicating that the commands were completed successfully.

```
SQLQuery6.sql - (...N5LVHQ\ENDER (56))* -> X
```

```
ALTER PROCEDURE gunluk
WITH ENCRYPTION
AS
SELECT COUNT (*) FROM tblSiparisDetay SD INNER JOIN tblSiparis S
ON SD.faturaKod = S.faturaKod
WHERE S.siparisTarih > GETDATE ()-1
AND S.siparisTarih < GETDATE ()+1
```

146 %

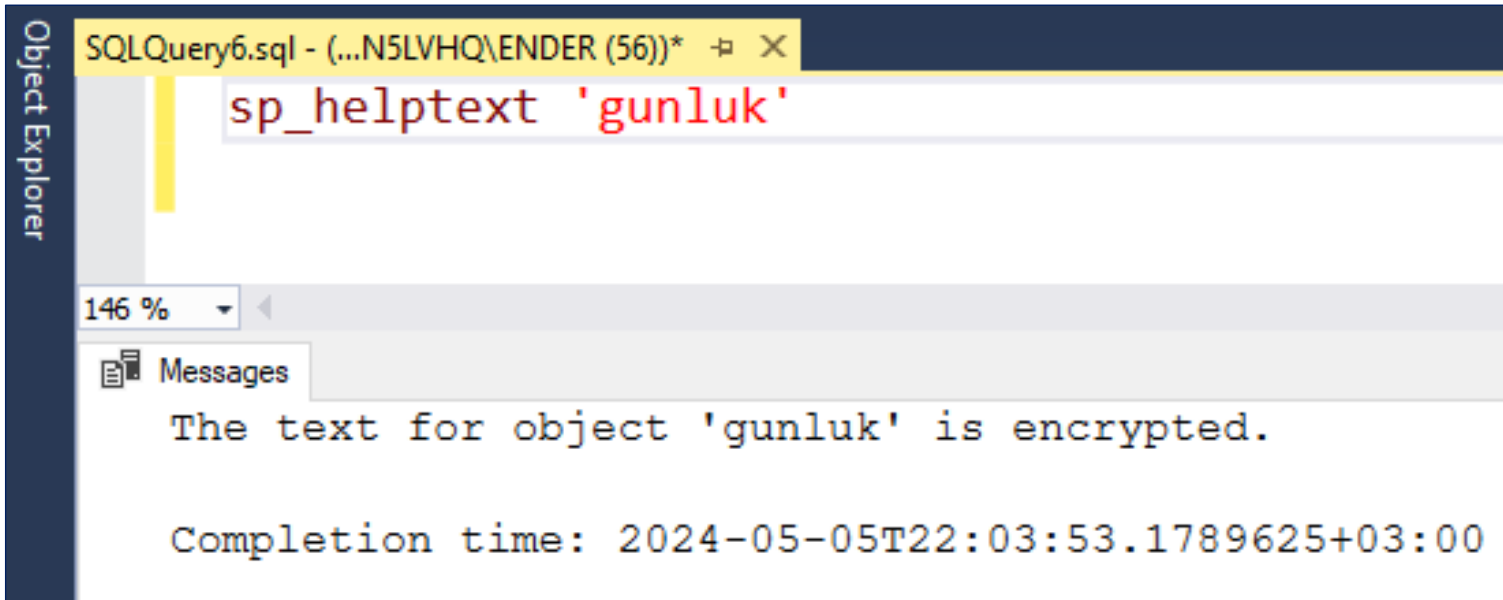
Messages

Commands completed successfully.

Completion time: 2024-05-05T22:02:26.7246566+03:00

10.4. SP'LERDE DEĞİŞİKLİK YAPMAK (DEVAM...)

sp_helptext 'gunluk' şeklinde çalıştırıp kodları görmek istediğimizde SP kodları görüntülenmeyecektir.



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the command `sp_helptext 'gunluk'` in a query window titled "SQLQuery6.sql - (...N5LVHQ\ENDER (56))*". The bottom pane, labeled "Messages", shows the error message: "The text for object 'gunluk' is encrypted." Below this, the completion time is listed as "2024-05-05T22:03:53.1789625+03:00". The zoom level is set to 146%.

```
SQLQuery6.sql - (...N5LVHQ\ENDER (56))* X
sp_helptext 'gunluk'

146 %
Messages
The text for object 'gunluk' is encrypted.

Completion time: 2024-05-05T22:03:53.1789625+03:00
```

10.5. SP'LERDE PARAMETRE KULLANMAK

Standart prosedürlerde olduğu gibi **SP'lerde de prosedürlere parametre gönderilerek bir sonuç üretilebilmektedir.** Yani **dışarıdan girdi parametreleri** ile değer alabilmektedirler. Parametrelili SP'lerin genel kullanım biçimi şöyledir.

```
CREATE PROCEDURE prosedur_ismi
```

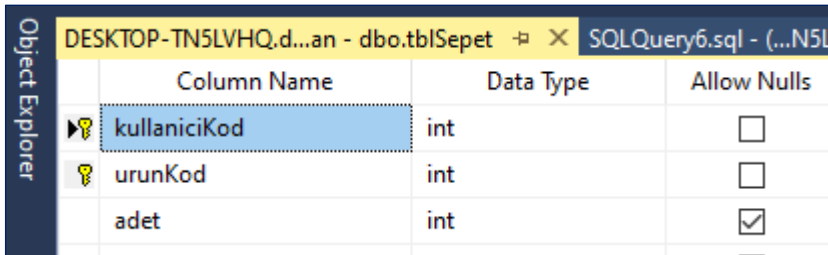
```
( @parametre_ismi veri_tipi, .....
```

```
AS
```

```
SQL ifadeleri
```

10.5. SP'LERDE PARAMETRE KULLANMAK (DEVAM)

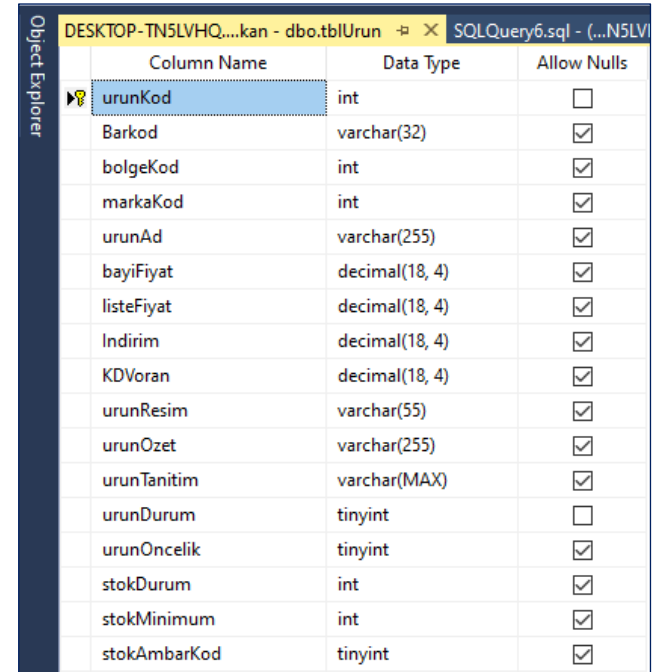
Örnek: Bir kullanıcı kodu alındığında o kullanıcıya ait sepette yer alan ürünleri görmek istiyoruz. Bu işlem için 'dukkân' VT'inde yer alan tblSepet ve tblUrun tabloları kullanılacaktır.



Object Explorer

DESKTOP-TN5LVHQ.d...an - dbo.tblSepet - SQLQuery6.sql - (...N5L)

	Column Name	Data Type	Allow Nulls
▶	kullaniciKod	int	<input type="checkbox"/>
▶	urunKod	int	<input type="checkbox"/>
	adet	int	<input checked="" type="checkbox"/>



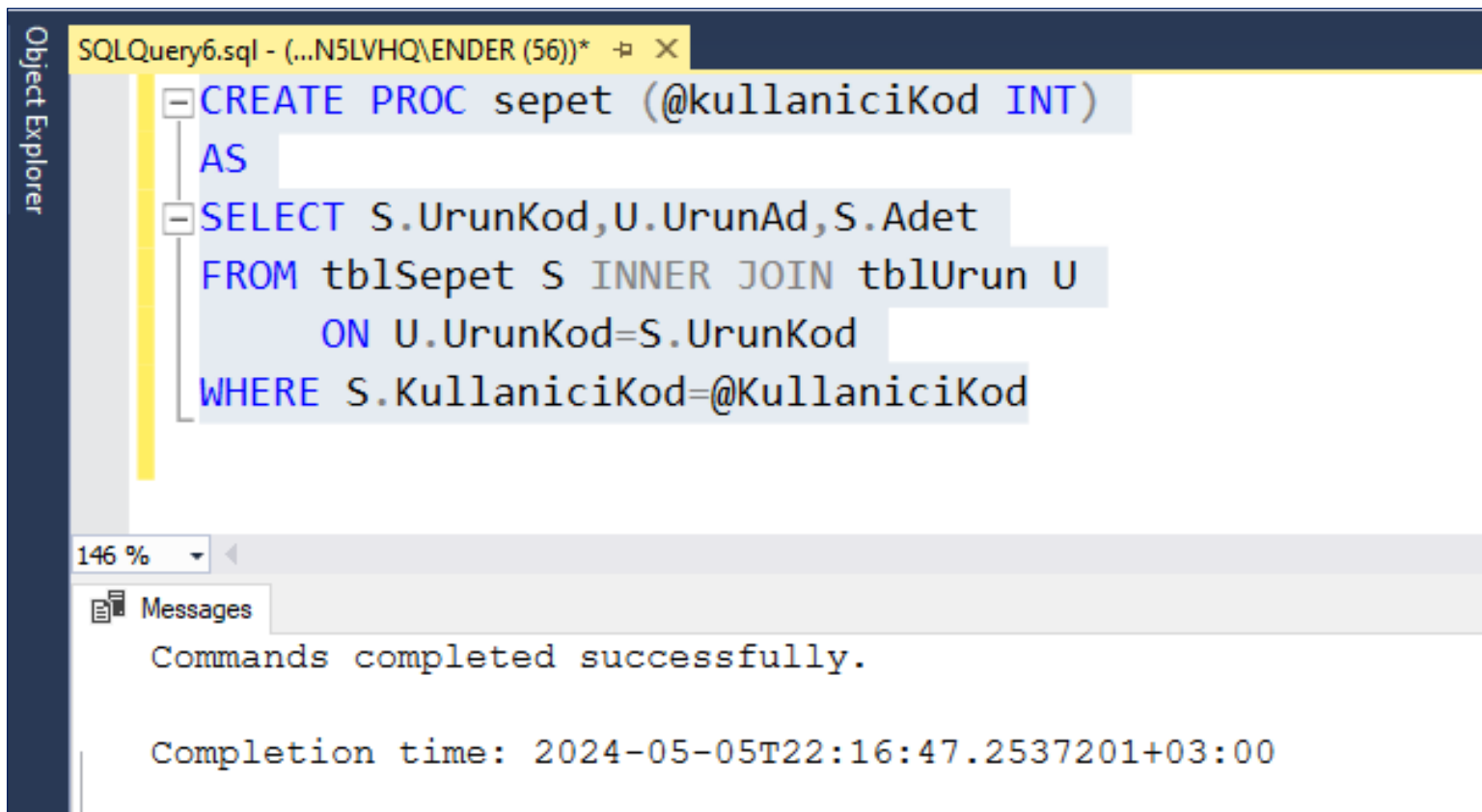
Object Explorer

DESKTOP-TN5LVHQ...kan - dbo.tblUrun - SQLQuery6.sql - (...N5LV)

	Column Name	Data Type	Allow Nulls
▶	urunKod	int	<input type="checkbox"/>
	Barkod	varchar(32)	<input checked="" type="checkbox"/>
	bolgeKod	int	<input checked="" type="checkbox"/>
	markaKod	int	<input checked="" type="checkbox"/>
	urunAd	varchar(255)	<input checked="" type="checkbox"/>
	bayiFiyat	decimal(18, 4)	<input checked="" type="checkbox"/>
	listeFiyat	decimal(18, 4)	<input checked="" type="checkbox"/>
	Indirim	decimal(18, 4)	<input checked="" type="checkbox"/>
	KDVoran	decimal(18, 4)	<input checked="" type="checkbox"/>
	urunResim	varchar(55)	<input checked="" type="checkbox"/>
	urunOzet	varchar(255)	<input checked="" type="checkbox"/>
	urunTanitim	varchar(MAX)	<input checked="" type="checkbox"/>
	urunDurum	tinyint	<input type="checkbox"/>
	urunOncelik	tinyint	<input checked="" type="checkbox"/>
	stokDurum	int	<input checked="" type="checkbox"/>
	stokMinimum	int	<input checked="" type="checkbox"/>
	stokAmbarKod	tinyint	<input checked="" type="checkbox"/>

10.5. SP'LERDE PARAMETRE KULLANMAK (DEVAM)

Buna göre yazılması gereken SP aşağıdaki gibi olacaktır.



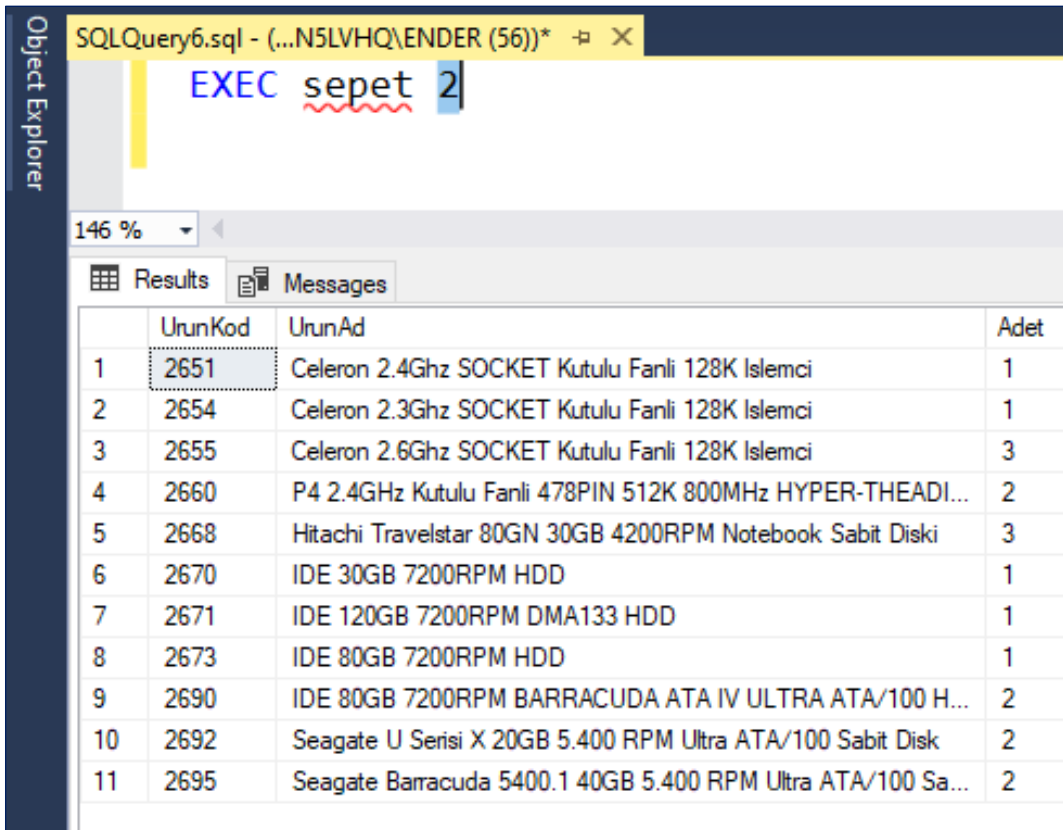
The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane shows a tree view with a minus sign next to the 'Queries and Functions' folder. The main window shows a SQL script in a text editor titled 'SQLQuery6.sql - (...N5LVHQ\ENDER (56))*'. The script defines a stored procedure named 'sepet' with a parameter '@kullaniciKod INT'. The procedure body consists of a SELECT statement that joins 'tblSepet' (aliased as S) and 'tblUrun' (aliased as U) on 'UrunKod', filtering by 'KullaniciKod = @KullaniciKod'. Below the script, the 'Messages' pane shows the output: 'Commands completed successfully.' and 'Completion time: 2024-05-05T22:16:47.2537201+03:00'.

```
SQLQuery6.sql - (...N5LVHQ\ENDER (56))*
--
CREATE PROC sepet (@kullaniciKod INT)
AS
SELECT S.UrunKod,U.UrunAd,S.Adet
FROM tblSepet S INNER JOIN tblUrun U
ON U.UrunKod=S.UrunKod
WHERE S.KullaniciKod=@KullaniciKod
--
146 %
Messages
Commands completed successfully.

Completion time: 2024-05-05T22:16:47.2537201+03:00
```

10.5. SP'LERDE PARAMETRE KULLANMAK (DEVAM)

Bu SP'nin çalıştırılması ise parametre verilerek aşağıdaki gibi olacaktır.



SQLQuery6.sql - (...N5LVHQ\ENDER (56))*

EXEC sepet 2

146 %

Results Messages

	UrunKod	UrunAd	Adet
1	2651	Celeron 2.4Ghz SOCKET Kutulu Fanli 128K Islemci	1
2	2654	Celeron 2.3Ghz SOCKET Kutulu Fanli 128K Islemci	1
3	2655	Celeron 2.6Ghz SOCKET Kutulu Fanli 128K Islemci	3
4	2660	P4 2.4GHz Kutulu Fanli 478PIN 512K 800MHz HYPER-THEADI...	2
5	2668	Hitachi Travelstar 80GN 30GB 4200RPM Notebook Sabit Diski	3
6	2670	IDE 30GB 7200RPM HDD	1
7	2671	IDE 120GB 7200RPM DMA133 HDD	1
8	2673	IDE 80GB 7200RPM HDD	1
9	2690	IDE 80GB 7200RPM BARRACUDA ATA IV ULTRA ATA/100 H...	2
10	2692	Seagate U Serisi X 20GB 5.400 RPM Ultra ATA/100 Sabit Disk	2
11	2695	Seagate Barracuda 5400.1 40GB 5.400 RPM Ultra ATA/100 Sa...	2

TEŞEKKÜRLER...

SORULARINIZ ?
