

# FONKSİYONLAR ve SAKLI YORDAMLAR (STORED PROCEDURES)

Öğretim Görevlisi A. Berika VAROL MALKOÇOĞLU

# İçindekiler

- Fonksiyonlar
- Saklı yordam nedir?
- Saklı Yordamlar
- Saklı Yordam örnekleri

# Fonksiyonlar

- Bir veya birden fazla parametre alır işler ve değer döndürür.
- Functionlar istenilen değer tipinde dönüş yapabilir. INT, VARCHAR değer döndürebileceğiniz gibi bir tablo da döndürebiliriz.
- Dışarıdan parametre alan bir view tanımlanamadığı için bu türden ihtiyaçlar için fonksiyon kullanılır.
- Saklı yordamları bir sorgunun parçası olarak kullanamayız ya da sorgulayamayız. Ancak view ve fonksiyonları sorgulayabiliriz.

# Syntax:

```
DELIMITER $$
```

```
CREATE FUNCTION function_name( param1, param2,... )
```

```
RETURNS datatype
```

```
[NOT] DETERMINISTIC
```

```
BEGIN
```

```
    -- statements
```





```
END
```

```
$$ DELIMITER ;
```

# Örnek

```
1 • use havayolu_otomasyonu_2;
2
3 DELIMITER $$
4 • CREATE FUNCTION id2pilotad(p_id int)
5 RETURNS varchar(50)
6 DETERMINISTIC
7 BEGIN
8     DECLARE pilot_ad VARCHAR(20);
9     select ad INTO pilot_ad from pilot where pilot_id=p_id;
10    RETURN pilot_ad;
11 END$$
12 DELIMITER ;
13
14 • SELECT pilot_id,id2pilotad(pilot_id) as 'pilot ad' FROM pilot;
15
```

<

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

	pilot_id	pilot ad
▶	1	Ahmet
	2	Ali
	3	Ali
	4	Ayşe

# Skaler tanımlı (tek değer döndüren) fonksiyonlar





- Skaler değerli fonksiyonlar bir tek değer döndüren fonksiyonlardır.
- Örnek `getdate()` bir skaler fonksiyondur, çünkü tek bir değer yani şu anın tarih ve saatini döndürür.
- Bazı durumlarda, buna benzer fonksiyonlara ihtiyaç duyulur.
- Örneğin bir müşterinin sepetinde kaç ürünün bulunduğu , kaç sipariş verdiğini döndüren fonksiyonlar tanımlanabilir.

# Örnek:

```
1 • USE eczane;
2 DELIMITER $$
3
4 • CREATE FUNCTION sepetUrunSayi(mno int)
5 RETURNS int
6 DETERMINISTIC
7 BEGIN
8     Declare urunSayi int;
9     SELECT sum(urun_adet) INTO urunSayi FROM musteri WHERE mno=musteri_no;
10    RETURN urunSayi;
11 END $$
12 DELIMITER ;
```

# Örnek:

14 • `select ad, soyad, sepetUrunSayi(musteri_no) from musteriler ;`



<			
Result Grid     Filter Rows: <input type="text"/>   Export:    Wrap Cell Content: 			
	ad	soyad	sepetUrunSayi(musteri_no)
▶	Aslı	Can	2
	Ayşe	Al	1
	Fatma	Saygın	1
	Emine	Yılmaz	1
	NULL	NULL	1
	ali	veli	4



# Örnek:

```
1 • use e_ticaret;  
2 DELIMITER $$  
3 • CREATE FUNCTION MusteriSeviye(s_tutar DECIMAL(10,2))  
4 RETURNS VARCHAR(20)  
5 DETERMINISTIC  
6 BEGIN  
7 DECLARE siparis_tutar VARCHAR(20);  
8  
9 IF s_tutar < 500 THEN  
10 SET siparis_tutar = 'SILVER';  
11 ELSEIF (s_tutar >= 500 AND  
12 s_tutar <= 5500) THEN  
13 SET siparis_tutar = 'GOLD';  
14 ELSEIF s_tutar > 5500 THEN  
15 SET siparis_tutar = 'PLATINUM';  
16 END IF;  
17 RETURN (siparis_tutar);  
18 END$$  
19 DELIMITER ;
```

```
23 • select m.ad, m.soyad, su.tutar, MusteriSeviye(tutar) from siparis_urun su  
24 inner join siparis s on su.siparis_id=s.id  
25 inner join musteriler m on s.musteri_id=m.id group by m.id;
```

Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content: 				
	ad	soyad	tutar	MusteriSeviye(tutar)
▶	aslı	kara	5604.19	PLATINUM
	utku	şen	5604.19	PLATINUM
	fatma	öz	5604.19	PLATINUM
	hasan	yılmaz	113.19	SILVER
	önder	özdemir	5423.19	GOLD
	kasım	güneş	442.19	SILVER
	halim	mutlu	15260.9	PLATINUM

# Saklı Yordamlar

- Veritabanında saklanan hazır derlenmiş sql kod bloklarıdır.
- Saklı yordamlar uygulamanın performansını yükseltir.
- Uygulama ve veritabanı sunucusu arasındaki trafiği azaltır.
- Saklı yordamlar tekrar kullanılabilir.
- Saklı yordamlar güvenlidir.

# Saklı Yordamlar

- Oluşturulan veya var olan saklı yordamlar dışarıdan parametre alabilirler.
- Bir kez yazılıp, tekrar tekrar kullanıldığı için program modüler bir yapıda geliştirilmiş olur.
- Otomatik devreye giremezler.
- Uygulama ya da script tarafından çağırılmaları gereklidir.

# Neden Kullanılır?

- Kodların yeniden kullanımı
  - Aynı veritabanını kullanan farklı uygulamalar tekrar aynı kodları yazmak yerine saklı yordamları kullanabilir.
- Daha kolay kodlama
  - Geliştirici sorgu ya da tabloların adını tam olarak bilmeden bir saklı yordamı çağırabilir.
  - Veritabanını daha kullanıcı dostu kılarlar.

# Saklı Yordamlar Yapısı

- CREATE PROCEDURE ya da kısaca CREATE PROC ifadesi ile yaratılır.

```
CREATE PROCEDURE procedure_name  
BEGIN  
    sql_statement;  
END;
```

- Saklı yordamı yürütme için;

```
CALL procedure_name;
```

# Saklı Yordamlar Örnek

```
CREATE PROCEDURE GetAllClients()  
BEGIN  
    SELECT * FROM Clients;  
END;  
  
CALL GetAllClients();
```

# Saklı Yordam İçerisinde Değişkenler Tanımlamak

- Saklı yordamlar içerisinde kullanabileceğimiz değişkenler tanımlayabiliriz.
- Değişken tanımlama DECLARE direktifi ile yapılır.

```
DECLARE degisken_adi veritipi(boyut) DEFAULT varsayilan_deger;
```

```
DECLARE totalBandwidth INT(20);
```

# Saklı Yordam İçerisinde Değişkene Değer Atamak

- Stored procedure içerisinde tanımlanmış bir değişkene iki yolla değer atanabilir.
- Sql cumlecisi ile;

```
SELECT count(*) INTO totalBandwidth FROM NetworkReport;
```

- Set direktifi ile;

```
SET totalBandwidth = 3412314124;
```



# Değişken Ömrü

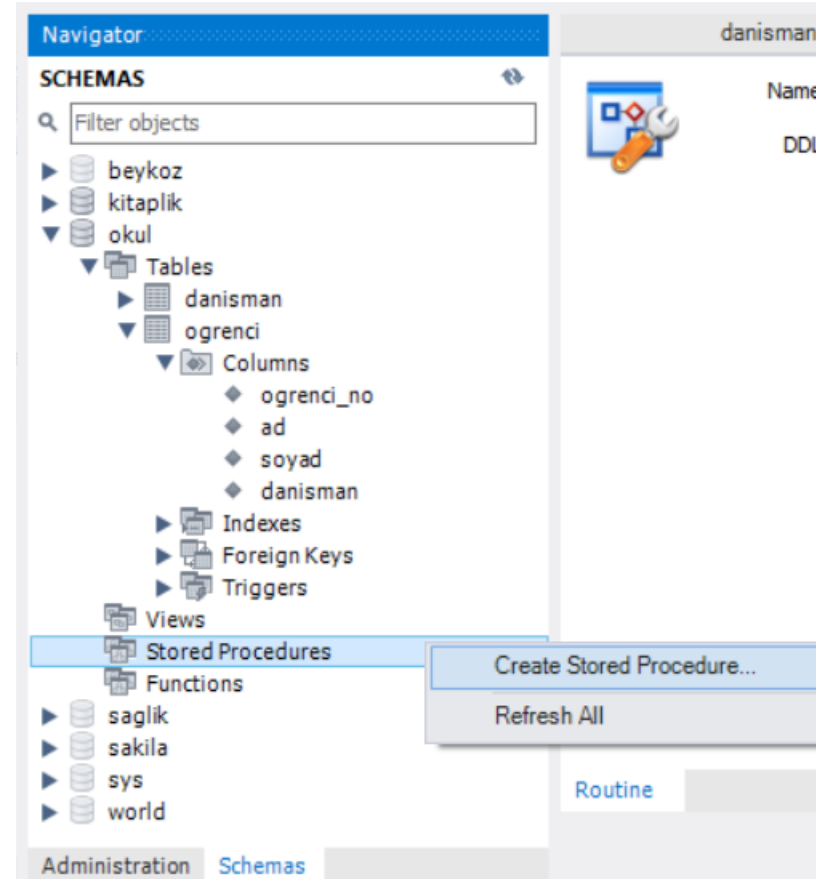
```
DELIMITER//  
CREATE PROCEDURE TestProcedure()  
BEGIN  
    DECLARE totalBandwidth int(20);  
    BEGIN  
        DECLARE totalHit int(20);  
    END  
END //  
DELIMITER;
```

# Saklı Yordamlar Örnek

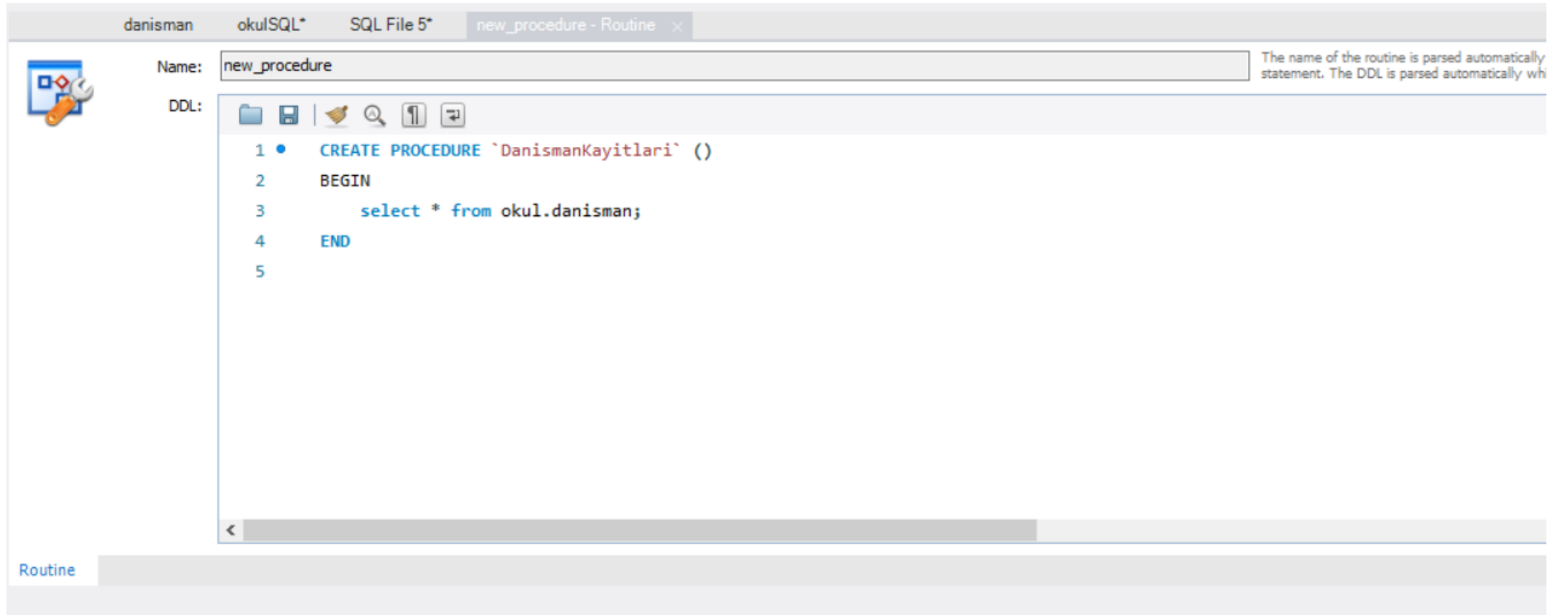
```
DECLARE benim_adim VARCHAR(50) DEFAULT '';  
SET benim_adim = 'berika';
```

```
DELIMITER //  
CREATE PROCEDURE SakliYordam()  
BEGIN  
    DECLARE benim_adim VARCHAR(50) DEFAULT '';  
    SET benim_adim = 'berika';  
    SELECT CHAR_LENGTH(benim_adim) AS Uzunluk;  
END//  
DELIMITER ;
```

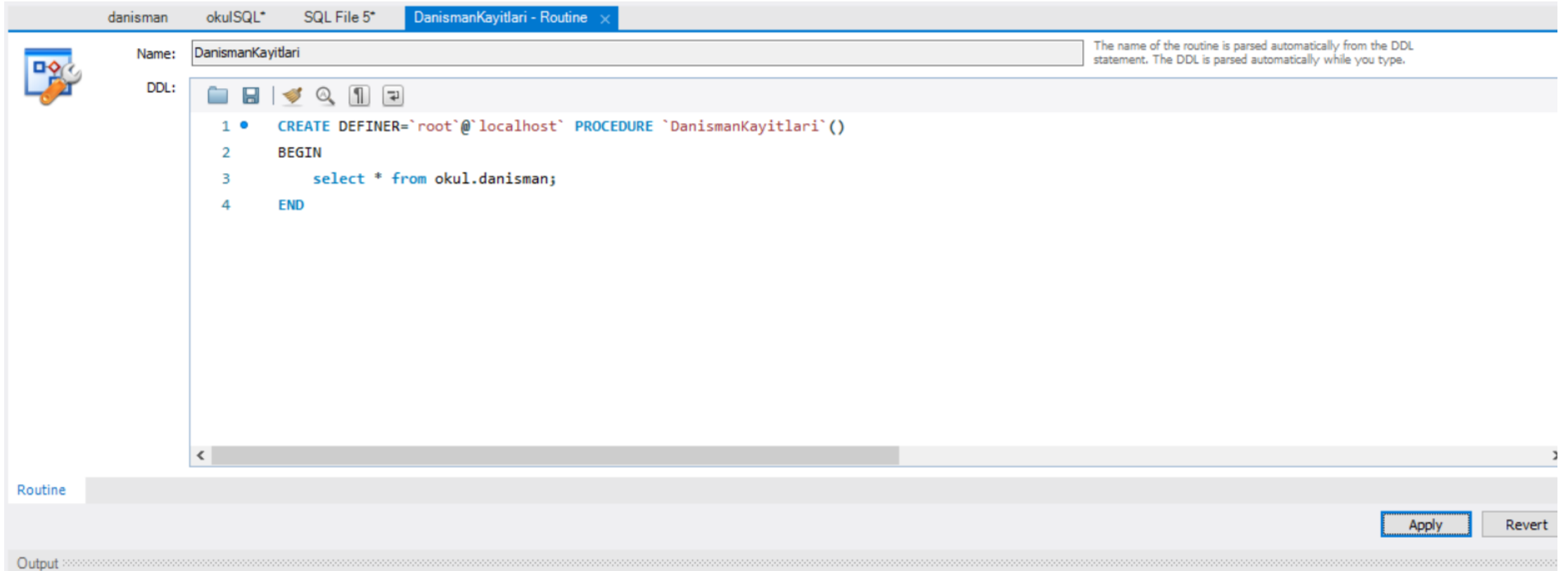
# Örnek



# Örnek



# Örnek

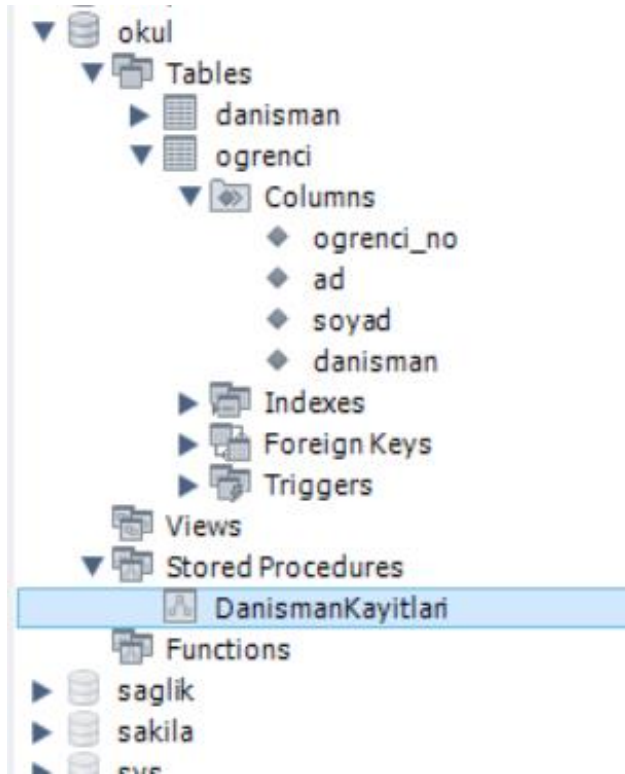


The screenshot displays a MySQL IDE window with the following components:

- Tab Bar:** Contains tabs for 'danisman', 'okulSQL\*', 'SQL File 5\*', and the active 'DanismanKayitlari - Routine'.
- Name Field:** Labeled 'Name:', it contains the text 'DanismanKayitlari'. A tooltip on the right states: 'The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.'
- DDL Field:** Labeled 'DDL:', it contains a SQL script:

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `DanismanKayitlari`()
2 BEGIN
3     select * from okul.danisman;
4 END
```
- Buttons:** 'Apply' and 'Revert' buttons are located at the bottom right of the DDL field.
- Output:** An 'Output' section is visible at the very bottom of the window.

# Örnek



48

49 • `CALL DanismanKayitlari;`

<

Result Grid | Filter Rows: | Export: | Wrap

	danisman_id	ad	soyad	departman
▶	1	Alican	ÇAN	BİLGİSAYAR PROG.
	2	Mehmet	GEZER	FİZİK
	3	Aslı	ŞENTÜRK	GEZER ATİK
	4	Ayşe	DAL	YAZILIM MÜHENDİSLİĞİ

# Örnek

```
51 • USE okul;  
52 • DROP procedure IF EXISTS OgrenciKayitlari;  
53  
54 DELIMITER $$  
55 • USE okul$$  
56 • CREATE PROCEDURE OgrenciKayitlari ()  
57 BEGIN  
58     select * from okul.ogrenci;  
59 END$$  
60  
61 DELIMITER ;  
62  
63  
64 • CALL OgrenciKayitlari;
```


< Result Grid | Filter Rows: | Export: | Wrap

	ogrenci_no	ad	soyad	danisman
▶	1000	Şeyma	ÇAĞLAR	1
	1001	Çiğdem	GEZER	3
	1002	Süleyman	ARSLAN	1

# Örnek

```
66
67 DELIMITER $$
68 ● USE okul$$
69 ● CREATE PROCEDURE Adinda_ey_GecenOgrenciler ()
70 BEGIN
71     SELECT * FROM okul.ogrenci WHERE ad LIKE '%ey%';
72 END$$
73
74 DELIMITER ;
75
76 ● CALL Adinda_ey_GecenOgrenciler;
```

<

Result Grid | Filter Rows:  | Export:  | Wrap Cell Co

	ogrenci_no	ad	soyad	danisman
▶	1000	Şeyma	ÇAĞLAR	1
	1002	Süleyman	ARSLAN	1



# Parametrelili Saklı Yordamlar

- Yarattığımız saklı yordamlar genellikle parametrelili olurlar.
- Parametre ile saklı yordama değer gönderebilir, değer alabiliriz.
- Mysql'de saklı yordam parametreleri 3 yöntem belirtecinden birini alabilir.
  - IN
  - OUT
  - INOUT

# IN Belirteci

- Parametreye sadece değer göndereceksek kullanırız.

```
CREATE PROCEDURE GetAge (IN p_personId INT(3))  
BEGIN  
    SELECT age FROM Persons WHERE personId = p_personId  
END
```

Call GetAge (345);

# OUT Belirteci

- Parametreye dışarıdan bir değişken gönderip saklı yordamın bu değişkene bir değer atamasını sağlarız.

```
CREATE PROCEDURE StudentTotal (IN s_name VARCHAR(45),  
OUT total INT)  
BEGIN  
    SELECT COUNT(*) INTO total FROM Student  
    WHERE name = s_name  
END
```

Call StudentTotal ('Ayşe' @total);

# INOUT Belirteci

- Hem saklı yordama değer gönderilebilir hem de saklı yordamdan bir değer alınabilir.

```
CREATE PROCEDURE StudentUpdate (INOUT id INT)
BEGIN
    UPDATE Student SET s_id=s_id+5 WHERE s_id=id
END;
```

SET @id=1;

Call StudentUpdate (@id);