

**T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



Veri Tabanı Yönetimi Laboratuvarı

Föy 4

15060007 Kübra Nur Saruhan

1. Fonksiyona girdileri 2 adet parametre ile gönderdik (parametre2%Parametre1) kontrolü ile 1. parametrenin 2. parametreye tam bölünüp bölünmediğini kontrol ettirdik. 50/5 , 50/7 ile fonksiyonu test ettik.

```
1 use vtys4hafta;
2 DELIMITER //
3 CREATE FUNCTION fn_BolunurMu(parametre1 INT, parametre2 INT)
4 RETURNS varchar(25)
5 BEGIN
6     IF (parametre2 % parametre1) = 0 THEN
7         RETURN 'Tam Bölünür';
8     ELSE
9         RETURN 'Tam Bölünmez';
10    END IF;
11 END//
12 DELIMITER ;
13
```

Output

#	Time	Action
✓ 1	02:02:08	use vtys4hafta
✓ 2	02:02:08	CREATE FUNCTION fn_BolunurMu(parametre1 INT, parametre2 INT) RETURNS varchar(25)

1 use vtys4hafta;	1 use vtys4hafta;
2 SELECT fn_BolunurMu(5,50);	2 SELECT fn_BolunurMu(7,50);
3	3

Result Grid	Filter Rows:
fn_BolunurMu(5,50)	
Tam Bölünür	

Result Grid	Filter Rows:
fn_BolunurMu(7,50)	
Tam Bölünmez	

2. Declare ile sonuç ve i adında 1 değerinde değişken tanımladık. While döngüsü ile parametreden gelen sayının faktöriyelini hesapladık .Return ile sonucu döndürdük.

```
1 use vtys4hafta;
2 DELIMITER //
3 CREATE FUNCTION faktoriyel(sayi INT)
4 RETURNS INT
5 BEGIN
6     DECLARE sonuc int DEFAULT 1;
7     DECLARE i int DEFAULT 1;
8     WHILE (i <= sayi) DO
9         SET sonuc = sonuc * i;
10        SET i = i + 1;
11    END WHILE;
12    return sonuc;
13 END//
14 DELIMITER ;
```

Output

#	Time	Action
✓ 1	02:13:15	use vtys4hafta
✓ 2	02:13:15	CREATE FUNCTION faktoriyel(sayi INT) RETURNS

Result Grid	Filter Rows:
faktoriyel(6)	
720	

3. ad, soyad, bolum ve maas sütunlarından 'calisanlar' isminde tablo oluřturup 3kiři ekledik

```
1 • use vtys4hafta;
2 • CREATE TABLE calisanlar (
3     ad VARCHAR(15) NULL ,
4     soyad VARCHAR(15) NULL,
5     bolum VARCHAR(25) NULL,
6     maas INT NULL
7 )
8
```

Output

Action Output

#	Time	Action
✓ 1	20:52:47	use vtys4hafta
✓ 2	20:52:47	CREATE TABLE calisanlar (ad VARC

```
1 • use vtys4hafta;
2 • INSERT INTO calisanlar (ad, soyad, bolum, maas)
3 VALUES('Kübra', 'Saruhan', 'muhendis', 2000),
4         ('Sümeýra', 'Aydın', 'ogretmen', 3000),
5         ('Elif', 'Yetkin', 'muhendis', 4000)
6
```

Output

Action Output

#	Time	Action
✓ 1	20:54:37	use vtys4hafta
✓ 2	20:54:37	INSERT INTO calisanlar (ad, soyad, bolum, maas) VALUES('Kübra

a) Bölüm bilgisinden çalışan personellerin maařına %25 zam yapan prosedür yazınız.

Zam adında bir prosedür oluřturarak muhendis çalışanlarının maařlarını $maas = maas * 1,25$ ile %25 zam arttırdık. İlk bařta girilen mühendis maařları 2000 iken 2500, 4000 iken 5000 řeklinde güncellendi.

```
1 use vtys4hafta;
2 DELIMITER //
3 CREATE PROCEDURE zam()
4 BEGIN
5     UPDATE calisanlar SET maas = maas*1.25
6     WHERE bolum = 'muhendis';
7 END //
8 DELIMITER ;
9 CALL zam();
10 SELECT * FROM calisanlar;
11
```

Result Grid

	ad	soyad	bolum	maas
▶	Kübra	Saruhan	muhendis	2500
	Sümeýra	Aydın	ogretmen	3000
	Elif	Yetkin	muhendis	5000

calisanlar 1 x

Output

Action Output

#	Time	Action
✓ 1	21:00:28	use vtys4hafta
✓ 2	21:00:28	CREATE PROCEDURE zam() BEGIN UPDATE calisanlar SET
✓ 3	21:00:28	CALL zam()
✓ 4	21:00:28	SELECT * FROM calisanlar LIMIT 0, 1000

b) Parametre olarak bölüm bilgisini alıp bu bölümdeki çalışanların maaşları toplamını döndüren prosedürü yazınız.

Parametre olarak gönderilen mühendis maaşlarının where ile kontrolü yaptırılarak sum(maas) olarak toplanarak değer döndürüldü.

```
1 use vtys4hafta;
2 DELIMITER //
3 CREATE PROCEDURE toplam(IN parametreBolum varchar(25))
4 BEGIN
5     SELECT SUM(maas) AS bolumMaasToplam FROM calisanlar WHERE bolum = parametreBolum ;
6 END //
7 DELIMITER ;
8 CALL toplam('muhendis');
9
```

Result Grid

	bolumMaasToplam
▶	7500

4. tOgrenci tablosu, ogrenciID (Primary Key), ad, soyad, dogumtarih, adres, telefon, bolumID, vizenotu, finalnotu şeklinde oluşturuldu ve veriler eklendi.

```
USE vtys4hafta ;
GO
CREATE TABLE tOgrenci (
    ogrenciID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    ad NVARCHAR(10) NULL,
    soyad NVARCHAR(10) NULL,
    dogumTarihi nvarchar(15) NULL,
    adres NVARCHAR(15) NULL,
    telefon NVARCHAR(11) NULL,
    bolumID INT NULL,
    vizeNotu INT NULL,
    finalNotu INT NULL
)

100 %
Messages
Commands completed successfully.
```

```
USE vtys4hafta
GO
INSERT INTO tOgrenci (ad, soyad, dogumTarihi, adres, telefon, bolumID, vizeNotu, finalNotu)
VALUES('Kübra', 'Saruhan', '1996-02-27', 'Samsun', '5555555555', 1, 80, 80),
      ('Nur', 'Ceng', '1990-01-20', 'Samsun', '5555555553', 1, 80, 80);

100 %
Messages
(2 rows affected)
```

```
sorgu4.sql - LAPTO...ELKG\saruhan (55) -> X
use vtys4hafta
go
DECLARE @ad nvarchar(10)
DECLARE @vizeNotu int
DECLARE tCursor cursor for
SELECT ad,vizeNotu FROM tOgrenci
OPEN tCursor
    FETCH NEXT FROM tCursor INTO @ad, @vizeNotu
    WHILE @@FETCH_STATUS = 0
    BEGIN
        UPDATE tOgrenci SET vizeNotu = vizeNotu + 5 WHERE ad LIKE '%a%' or ad LIKE '%i%';
        UPDATE tOgrenci SET vizeNotu = vizeNotu - 5 WHERE ad NOT LIKE '%a%' and ad NOT LIKE '%i%';
        FETCH NEXT FROM tCursor INTO @ad, @vizeNotu
    END
CLOSE tCursor
DEALLOCATE tCursor
SELECT * FROM tOgrenci
```

	ogrenciID	ad	soyad	dogumTarihi	adres	telefon	bolumID	vizeNotu	finalNotu
1	1	Kübra	Saruhan	1996-02-27	Samsun	5555555555	1	85	80
2	2	Nur	Ceng	1990-01-20	Samsun	5555555553	1	75	80

Where koşulu ile ‘%a%’ içinde a geçen , ‘%i%’ içinde i geçen kelimeleri bulmamızı sağlayarak vize notuna 5 puan ekleme, bu harflerin geçmediği öğrencilerin vize notundan 5 puan eksiltme kontrolü yaptırıldı. Declare ile tanımlamalar yapıldı. Open ile cursoru açtık. Fetch next From ile verileri fetch_status 0 olana kadar çağırdık. Close ile kapatıp deallocate bellekten cursoru sildik.