



YALOVA ÜNİVERSİTESİ  
university of yalova

Öğr. Gör. Cansu AYVAZ GÜVEN

# VERİTABANI-II

Değişken Tanımlama Ve Akış Kontrol Deyimleri

# Değişken Nedir?

- Değişkenler, programın veya kodların icra süresince belirli bir değer tutan ve istenilirse bu değer değiştirilebilen programlama elemanlarıdır.
- Değişkenleri kullanabilmek için öncelikle tanımlamak gerekmektedir.

# Değişken Tanımlama

- Değişken isimlerinin başında @ sembolü bulunur. @ad, @soyad, @tarih değişken isimlerine birer örnektir.
- Değişken tanımlamak için **DECLARE** deyimi kullanılır.
- Genel ifade

**DECLARE** @degisken\_adi **veritipi**  
şeklinde tanımlanır.

# Değişken Tanımlama

- DECLARE @ad nvarchar(25)
- DECLARE @sayi int

Tek bir satırda farklı tiplerde birden fazla değişken tanımlaması yapılabilir.

- DECLARE @numara int , @borc Money

# Değişkene Değer Atama

- **SET** @degiskenAdi = atanacakDeger
  - **DECLARE** @sayi1 **INT**, @sayi2 **INT**, @toplam **INT**  
**SET** @sayi1=16  
**SET** @sayi2=37  
**SET** @toplam=@sayi1+@sayi2

Değişkenlere atanacak değer sabit olabileceği gibi diğer değişkenlerden de elde edilebilir.

- @sayi = 10 şeklinde atama yapılamaz, hata verir.

# Değişkene Değer Atama

- Değişkene atanacak değer bir sorgudan dönen değer de olabilir. Bu durumda SELECT ifadesi kullanılmalıdır.

```
DECLARE @EnyuksekMaas Money
```

```
SELECT @EnyuksekMaas=MAX(Maas) FROM  
calisanlar
```

! SELECT ifadesi ile sabit değer ataması da yapılabilir.

# Sistem Fonksiyonları

- SQL server'da önceden tanımlı pek çok değişken vardır. Bunlardan iki tanesinden bahsedelim.
- @@error : sql deyimi hata vermeden çalışırsa değeri sıfır, hata verirse sıfırdan farklı bir değer alır.
- @@rowcount : Bir update, delete, insert deyimi çalıştığında, kaç tane kaydın etkilendiğini bize bildirir.

# Yazı Yazdırma

- Yazı yazdırma *PRINT* deyimi kullanılır.
- Örnek:

```
PRINT 'Ali Coşkun'
```

```
PRINT 5
```

```
PRINT 'HATA'
```

```
PRINT @@error
```



# Yazı Yazdırma

- `DECLARE @sayi1 INT, @sayi2 INT, @toplam INT`  
`SET @sayi1=16`  
`SET @sayi2=37`  
`SET @toplam=@sayi1+@sayi2`  
`PRINT @toplam`
- `DECLARE @EnyukseMaas Money`  
`SELECT @EnyukseMaas=MAX(Maas) FROM calisanlar`  
`PRINT @EnyukseMaas`

# Akış Kontrol Deyimleri

CASE

IF .... ELSE ....

WHILE .....

BREAK

CONTINUE

# CASE deyimi

- Case deyimi SELECT, UPDATE, DELETE, SET, IN, WHERE, ORDER BY gibi deyimlerin içinde kullanılır.
- Sql içinde belirli durumlara göre farklı değerler üretmek istiyorsak CASE deyimi kullanırız. İki farklı yazım şekli vardır.

# Basit CASE yapısı

## CASE GirişDenklemi

WHEN değer1 THEN sonuç1

WHEN değer2 THEN sonuç2

....

....

ELSE

END

# Örnek

```
DECLARE @sayi INT=5
SELECT
CASE @sayi
    WHEN 1 THEN 'Zayıf'
    WHEN 2 THEN 'Geçer'
    WHEN 3 THEN 'Orta'
    WHEN 4 THEN 'İyi'
    WHEN 5 THEN 'Pekiyi'
ELSE 'Aralık Dışı Değer'
END
```

# İkinci CASE yazımı

CASE

WHEN koşul1 THEN sonuç1

WHEN koşul2 THEN sonuç2

.....

ELSE sonuç

END

# Örnek

```
DECLARE @notu INT=50
SELECT
CASE
WHEN @notu>=0 AND @notu<50 THEN 'Kaldı'
WHEN @notu>=50 AND @notu<=100 THEN 'Geçti'
ELSE 'Aralık Dışı Not değeri'
END
```

Örnek: Çalışanlar tablosunda Medeni Durum değeri 1 olanlar için Bekar, 2 olanlar için Evli, diğer değerler için Bilinmiyor yazdıran sql kodları

```
SELECT Ad, Soyad,  
CASE  
WHEN MedeniDurum=1 THEN 'Bekar'  
WHEN MedeniDurum=2 THEN 'Evli'  
ELSE 'Bilinmiyor'  
END AS 'Medeni Durum'  
FROM calisanlar
```



# Örnek

- Çalışanlar tablosunda maaşı 2500 den küçük olan kayıtlar için ‘az maas’, 4000 den büyük olanlar için ‘çok maas’, bu değerlerin dışındaki değerler için ‘normal maas’
- BolumId 1 ise ‘ARGE’, 2 ise ‘Yazılım’, 5 ise ‘İdare’
- BolumId 1 ise %13, 2 ise %11 zam yapan sql ifadesini case yapısını kullanarak yazınız.

```
- SELECT *,  
CASE  
    WHEN maas<2500 THEN 'maaş Az'  
    WHEN maas>4000 THEN 'maaş çok'  
ELSE 'Maaş Normal'  
END AS MaasDurumu,  
CASE BoluId  
    WHEN 1 THEN 'ARGE'  
    WHEN 2 THEN 'YAZILIM'  
    WHEN 5 THEN 'İDARE'  
END as BoluAdi,  
CASE BoluId  
    WHEN 1 THEN maas*1.3  
    WHEN 2 THEN maas*1.1  
ELSE maas  
END as zamliMaas  
FROM calisanlar
```

# Ödev

- Öğrenci tablosundaki her bir öğrencinin öğrenci Numarasını, adını, soyadını ,ders adını ve 100 lük notuna göre harf notunu döndüren sorguyu yazınız. (100 lük notu harf notuna dönüştürürken case yapısı kullanınız.)

Ogrenci	
🔑	OgrenciNo
	Ad
	Soyad
	DersAdi
	Notu

# IF... ELSE... deyimi

**IF** koşul

Sql deyimi

Koşul doğru ise devamındaki sql deyimi çalışır. Yanlış ise çalışmaz.

**IF** koşul

Sql deyimi (1)

**ELSE**

Sql deyimi (2)

Koşul doğru ise 1. sql deyimi, yanlış ise 2. sql deyimi çalışır.

Eğer sql deyimi birden fazla satırdan oluşursa BEGIN .... END arasında yazılır.




Örnek: Çalışanlar tablosunda Ortalama maaş 3000'den çok ise “maaş ortalaması yüksek”, az ise “maaş ortalaması düşük” yazsın

```
IF (SELECT AVG(Maas) FROM calisanlar) > 3000  
PRINT 'Maaş ortalaması yüksek'  
ELSE  
PRINT 'Maaş ortalaması düşük'
```

Örnek: 5 nolu personelin maaşı ortalama maaştan düşük ise “az maaş”, değilse “çok maaş” yazdıran sql ifadesi

```
IF (SELECT Maas FROM calisanlar WHERE  
CalisanId=5) >  
(SELECT AVG(Maas) FROM calisanlar)  
PRINT 'çok maaş'  
ELSE  
PRINT 'az maaş'
```

# WHILE döngüsü

- While yapısı bir sql deyiminin tekrarlanmasını sağlar. Koşul yanlış oluncaya kadar döngü çalışır. 
- BREAK : döngüden koşul sağlanmadan yani erken çıkılmasını sağlar. 
- CONTINUE : devamındaki satırlar işlenmeden döngünün ilk satırına dönülmesini sağlar. 

# Genel ifade

**Veya**

**WHILE** koşul

Sql deyimi

**WHILE** koşul

**BEGIN**

Sql deyimleri

**END**

**Veya**

**WHILE** koşul

**BEGIN**

Sql deyimi

Koşula bağlı **BREAK** deyimi

Sql deyimi

Koşula bağlı **CONTINUE**  
deyimi

Sql deyimi

**END**



Örnek:1 den 12 ye kadar sayıları yazdıralım.

```
DECLARE @sayi int
SELECT @sayi=1
WHILE @sayi<=12
BEGIN
PRINT @sayi
SELECT @sayi = @sayi + 1
END
```



Örnek :1'den 30 kadar döngüyü çalıştıralım. 1'den 24'e kadar yazdıralım. 25 olunca döngüden çıkalım.

```
DECLARE @sayi int
SELECT @sayi =1
WHILE @sayi <=30
BEGIN
PRINT @sayi
IF @sayi = 24
BREAK
SELECT @sayi=@sayi+1
END
```



Örnek : 1den 30 a kadar olan sayıları yazdıralım. Ancak 15-20 arası yazılmasın!

```
DECLARE @sayi int
SELECT @sayi=0
WHILE @sayi < 30
BEGIN
SELECT @sayi=@sayi+1
IF @sayi Between 15 And 20
CONTINUE
PRINT @sayi
END
```

