

# KÜRSÖRLER devam

## Kürsörde iş akışı

- kürsörü oluştur
- oluşan kürsörü aç
- istenilen kayıda konumlan
- sql kodlarını çalıştır
- kürsörü kapat
- kürsörü yok et

## Kürsörü oluşturma

DECLARE ...1... cursor for ...2...

Veya

DECLARE ...1... SCROLL CURSOR  
FOR ...2...

1 nolu yere kürsörün adı

2 nolu yere ise kürsörün içinde dolaşacağı  
select sorgusu

Scroll kelimesi tanımda yok ise, kürsör  
sadece ileri hareket yapabilir.

Scroll kelimesi tanımda var ise, kürsör  
hertürlü hareketi yapabilir. (ileri, geri, ilk  
kayıda git, son kayıda git, n. kayıda git vs.)

**Kürsörü açmak, kapatmak ve yok etmek**

OPEN kürsörAdı --- açma

CLOSE kürsörAdı --- kapatma

DEALLOCATE kürsörAdı -- yok etme

**Kürsörü hareket etmek**

Aşağıdaki şekilde yazılır.

FETCH ....1... FROM ...2... INTO ...3...

1 nolu yere hareketin şekli:

NEXT : bir sonraki kayıta git

Aşağıdakiler sadece scroll tanımlı kürsörlerde çalışır.

PRIOR : önceki kayıta git

FIRST : ilk kayıta git

LAST :son kayıta git

ABSOLUTE n : n. kayıta git.

RELATIVE n : mevcut bulunulan kayıttan itibaren n. kayıt ileri yada geri git.

Geri için n negatif, ileri için n pozitif tamsayıdır.

Örnek RELATIVE -5 : 5 kayıt geri

RELATIVE 7 : 7 kayıt ileri git

2 nolu yere: kürsörün adı

3 nolu yere ise , kürsör tanımındaki select deyimine bağlı olarak, bulunulan yerdeki bilgilerin aktarılacağı değişkenler bulunur.

@@FETCH\_STATUS : hareket esnasında, hareket başarılı olmuş ise değeri sıfır, başarısız olmuş, veya son kayıta iken ileri

gitmek istediğimizde değeri sıfırdan farklı bir sayı olur.

Örnek:

Kürsörleri kullanarak personel tablosundaki 4 nolu kayıta bulunan personelin ad ve soyadını yazdıralım.

```
DECLARE @ad varchar(20), @soyad  
varchar(20)
```

```
DECLARE PersKursor SCROLL  
CURSOR FOR select ad, soyad from  
personel
```

```
OPEN PersKursor
```

```
FETCH absolute 4 FROM PersKursor  
INTO @ad, @soyad
```

```
PRINT @ad + ' ' + @soyad
```

```
CLOSE PersKursor
```

## DEALLOCATE PersKursor

Örnek:

Personel tablosunu 1. kayıttan sonuncu kayıta kadar ad, soyad değerlerini yazdırınız.

```
DECLARE @ad varchar(20), @soyad  
varchar(20)
```

```
DECLARE PersKursor CURSOR FOR  
select ad, soyad from personel
```

```
OPEN PersKursor
```

```
FETCH next FROM PersKursor INTO  
@ad, @soyad
```

```
WHILE @@fetch_status = 0
```

```
Begin
```

```
    PRINT @ad+' '+@soyad
```

```
    FETCH next FROM PersKursor INTO  
    @ad, @soyad
```

End

Close PersKursor

DEALLOCATE PersKursor

Örnek:

Personel tablosundaki kızların isimlerini yazdırınız. (cinsiyet =2)

Küresörün dolaşacağı sorgu:

```
SELECT ad, soyad from personel where  
cinsiyet =2
```

```
DECLARE @ad varchar(20), @soyad  
varchar(20)
```

```
DECLARE Pers_Kursor CURSOR FOR  
select ad, soyad from personel where  
cinsiyet = 2
```

OPEN Pers\_Kursor

```
FETCH next FROM Pers_Kursor INTO  
@ad, @soyad
```

```
While @@fetch_status=0  
Begin  
    PRINT @ad + ' ' +@soyad  
    FETCH next FROM Pers_Kursor  
INTO @ad, @soyad  
End
```

```
CLOSE Pers_Kursor  
DEALLOCATE Pers_Kursor
```

## **STORED PROCEDURE (SAKLI YORDAMLAR)**

Birden fazla işlemi, paketlenmiş bir halde, bir tek komut ile çalıştırmamız gerektiğinde stored procedure'leri kullanırız. İşlemlerden kastedilen Transact SQL ile yapılan her şeydir.

Network trafiğini azaltır, çünkü tüm işlemler sunucuda gerçekleşir, kullanıcı bilgisayara sadece sonuç gider.

Neler yapılabilir? Yapısı nasıl dır ?

Örnek c++ prosedürü yazalım

```
Void hesapla (int a, int b)
{
    b=a*2
}
```

Burada : hesapla ; prosedürün adı  
a , b : parametreler

- parametre bulunabilir
- parametresiz olabilir.
- İstendiğinde parametreden veri geri dönebilir.
- Liste çıktısı alınabilir.
- Her türlü sql deyimi çalışabilir.



Uygulama:

<http://uygulama.kilicaslan.nom.tr>