

## ORACLE AKIŞ KONTROL DEYİMLERİ

### IF...THEN...ELSE.. Deyimi

#### Genel ifadeler

IF koşul THEN

    Komutlar

END IF;

IF koşul THEN

    Komutlar

ELSE

    Komutlar

END IF;

IF koşul THEN

    Komutlar

ELSIF koşul THEN

    Komutlar

ELSE

    Komutlar

END IF;

```
declare
  d_departman number;
  d_sicilno char(10):='2053';
begin
  select departman into d_departman from
  personel where sicilno=d_sicilno;
  if d_departman=5 then
    dbms_output.put_line('5 nolu departmanda');
  else
    dbms_output.put_line('diğer departmanlarda');
  end if;
end;
```

yukarıda if deyimi uygulaması yapıldı.

**Yukarıdaki koda elsif ekleyelim.**

```
declare
  d_departman number;
  d_sicilno char(10):='2053';
begin
  select departman into d_departman from
  personel where sicilno=d_sicilno;
  if d_departman=5 then
    dbms_output.put_line('5 nolu departmanda');
```

```
elsif d_deptman=10 then
  dbms_output.put_line('10 nolu departmanda');
else
  dbms_output.put_line('diğer departmanlarda');
end if;
end;
```

## FOR DÖNGÜSÜ

### Yapısı

```
FOR değişken IN deger1..deger2
LOOP
  Komutlar
END LOOP;
```

Yukarıdaki kodda değişken değeri deger1den deger2'ye kadar birer artar.

### Örnek:

For döngüsü ile 1 den 5'e kadar sayıları sırası ile yazdıralım.

```
declare
  i number;
begin
```

```
for i in 1..5
loop
  dbms_output.put(i || ', ');
end loop;
dbms_output.put_line("");
end;
```

for döngüsünün geriye doğru sayması

```
FOR değişken IN REVERSE deger1..deger2
LOOP
  Komutlar
END LOOP;
```

Yukarı kodda deger2 den deger1 'e kadar geriye doğru sayılır.

Yukarıdaki örneği geriye saydıralım.

```
declare
  i number;
begin
  for i in reverse 1..5
  loop
    dbms_output.put(i || ', ');
```

```
end loop;  
  dbms_output.put_line("");  
end;
```

if ve for döngüsü karışık bir örnek

```
declare  
  i number;  
begin  
  for i in reverse 1..5  
  loop  
    if i=1 then  
      dbms_output.put_line(i);  
    else  
      dbms_output.put(i || ', ');  
    end if;  
  end loop;  
end;
```

## GOTO Deyimi

GOTO deyimi, program içinde akışın tanımlanan bir etikete(LABEL) gitmesini sağlar. Etiket <<etiketAdi>> şeklinde tanımlanır.

Kullanımı aşağıdaki gibidir.

begin

.....

<<etiketAdi>>

.....

GOTO etiketAdi

.....

end;

bu deyim kullanılarak döngüden çıkılabilir.

Örnek:

For döngüsü 1 den 10'a kadar saysın, değişken değeri 5 olunca goto deyimini ile döngüden çıkılsın.

```
declare
```

```
  i number;
```

```
begin
```

```
  for i in 1..10
```

```
  loop
```

```
    dbms_output.put_line(i);
```

```
    if i=5 then
```

```
      goto a;
```

```
    end if;
```

```
  end loop;
```

```
dbms_output.put_line('Bu satır atlandı');
```

```
<<a>>
```

```
dbms_output.put_line('Döğüden çıkıldı');  
end;
```

## LOOP Döngüsü

Bu döngüden GOTO ve EXIT deyimleri ile çıkılır.

## LOOP

Komutlar

```
END LOOP;
```

## Örnek:

1 den 5'e kadar loop döngüsü ile sayıları yazdıralım.

## Çözüm:

```
declare
```

```
  i number(5);
```

```
begin
```

```
  i:=1;
```

```
  loop
```

```
    dbms_output.put_line(i);
    if i=5 then
        exit;
    end if;
    i:=i+1;
end loop;
end;
```

## WHILE Döngüsü

Koşul doğru olduğu sürece dönen bir döngüdür.

WHILE koşul

LOOP

    Komutlar

END LOOP;

declare

    i number:=1;

begin

    while i<=5

    loop

        dbms\_output.put\_line(i);

        i:=i+1;

    end loop;



end;

## EXIT deyimi

Döngüden çıkmayı sağlar.

EXIT veya

EXIT WHEN koşul;

Koşul doğru olduğunda çıkış gerçekleşir.

EXIT WHEN i=5;

i değeri 5 olduğunda döngüde çıkar.

## KAYIT VERİTİPİ (ROWTYPE)

Bu veritipi tipi ile tanımlanmış değişkene, tablonun bir satırı atanabilir.

Tanımı

DeğişkenAdi tabloAdi%ROWTYPE;

p\_personel personel%ROWTYPE;

p\_personel isimli değişkene, personel tablosunun herhangi bir satırı atanabilir. Bu değişkenin

kolonları vardır. Değişkenin kolonları personel tablosunun kolonları ile aynıdır.

p\_personel.sicilno  
p\_personel.ad                      şeklinde kullanılacak.

Örnek:

2053 nolu personelin tüm bilgilerini p\_personel değişkenine atayınız ve bu değişkenden ad ve soyad bilgisini yazdırınız.

Çözüm:

```
declare
  p_personel  personel%rowtype;
begin
  select * into p_personel from personel where
  sicilno='2053';

  dbms_output.put_line(p_personel.ad || ' ' ||
  p_personel.soyad);
end;
```

## Kürsör

SQL%ISOPEN : kursor açılmı bilgisi bulunur.

SQL%ROWCOUNT: sql ile işlem gören kayıt sayısı bulunur.

SQL%FOUND : sql sonucu en az bir kayıt bulunursa değeri true'dur

SQL%NOTFOUND: sql sonucu hiçbir kayıt bulunamaz ise değeri true'dur.

Tanımlama:

declare

CURSOR kursorAdi IS sqlDeyimi

KayıttipiDeğişken kursorAdi%ROWTYPE;

Begin

    Open kursorAdi

    LOOP

        fetch kursorAdi into kayıtTipiDeğişken

        exit when kursorAdi%NOTFOUND;

        sql komutları

    end loop

end;

close kursorAdi

Örnek:

Kursorü kullanarak personel tablosundaki tüm ad ve soyadları yazdırınız.

```
declare
  cursor p_kursor is select * from personel;
  p_personel p_kursor%rowtype;
begin
  open p_kursor;
  loop
    fetch p_kursor into p_personel;
    exit when p_kursor%notfound;
    dbms_output.put_line(p_personel.ad||' '||
p_personel.soyad);
  end loop;
  close p_kursor;
end;
```

## Kullanıcı Tanımlı paketler

PL/SQL de pek çok hazır fonksiyon, prosedür vardır. Ayrıca belli amaçla bir araya getirilen komutlardan oluşan paketler vardır.

Paketler kullanıcı tarafından oluşturulabilir.  
Paketler ile nesne tabanlı programlama yapılabilir.

Paket iki blümden oluşur.  
Birincisi, paketin tanımı bölümü,

## İkincisi gövde bölümü.

### Paket Tanımı Bölümü:

Bu bölümde pakette bulunan değişkenler, fonksiyonlar, prosedürler tanımlanır.

#### Yapısı

```
create or replace package PaketAdi is
    Tanımlamalar
end paketAdi;
```

### Paket Gövdesi:

Paket tanımlamasında tanımlanan nesnelere haricinde, paketle ilgili bütün kodların bulunduğu bölüm paket gövdesidir.

#### Yapısı:

```
Create or replace package body paketAdi is
    Komutlar
End paketAdi;
```

### Örnek:

```
create or replace package personel_paket is
    procedure toplam_maas;
    procedure personel_maas(p_sicilno char);
    p_maas number(15,2);
```

```
p_personel personel%rowtype;  
procedure Personel_kayit(p_sicilno char);  
end personel_paket;
```

```
create or replace package body personel_paket is  
  procedure toplam_maas is  
  begin  
    select sum(maas) into p_maas from personel;  
  end;
```

```
  procedure personel_maas(p_sicilno char)  
  is  
  begin  
    select maas into p_maas from personel where  
  sicilno=p_sicilno;  
  end;
```

```
  procedure Personel_kayit(p_sicilno char) is  
  begin  
    select * into p_personel from personel where  
  sicilno=p_sicilno;  
  end;
```

```
end personel_paket;
```

yukarıdaki kodu kullanalım:

begin

```
personel_paket.toplam_maas;  
dbms_output.put_line(personel_paket.p_maas);
```

```
personel_paket.personel_maas('2053');  
dbms_output.put_line(personel_paket.p_maas);
```

```
personel_paket.Personel_kayit('2053');  
dbms_output.put_line(  
personel_paket.p_personel.ad||'  
||personel_paket.p_personel.maas);
```

end;