

ORACLE'DA KÜRSÖRLER

Gerekli sistem değişkenleri

SQL%ISOPEN : kursor açıksa değeri true, kapalı ise değeri false

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

SQL%FOUND : sql işlemi sonucu, en az bir kayıt bulunursa değeri true, değilse false'dur.

SQL%NOTFOUND : sql sonucu hiçbir kayıt dönmez ise değeri true, döner ise değeri false'dur.

Genel kursor kullanım kalıbı

declare

 CURSOR kursorAdi IS selectDeyimi;

 kayıtTipiDeğişken kursorAdi%ROWTYPE;

begin

 OPEN kursorAdi;

 LOOP

 FETCH kursorAdi INTO kayıtTipiDeğişken;

 EXIT WHEN kursorAdi%NOTFOUND;

 Komutlar

 END LOOP;

```
CLOSE kursorAdi;  
end;
```

Aşamaları metin olarak yazalım.

1-Kursorü tanımla

2-Kursorün her kayıtının aktarılacağı bir değişken tanımla

3-Kursörü aç

4-Döngü içinde,

a. Bir sonraki kayıda git

b. Son kayıda erişilmiş ise döngüden çık

c. Kursorde fayadalanacak kodları yaz

5-Kursorü kapat

Yukarıdaki kodda bulunan EXIT deyimini açıklayalım.

EXIT deyimi döngüden çıkmayı sağlar. İki çeşit kullanımını vardır.

EXIT veya

EXIT WHEN koşul

burada koşul doğru ise döngüden çıkılır. Tek başına EXIT kelimesi IF deyimi ile kullanılabilir.

Yukarıdaki metindeki IF ile yazarsak

.....

```
IF kursorAdi%NOTFOUND THEN
    EXIT
END IF
```

.....

Örnek:

Kursorü kullanarak personel tablosundaki, personelin sicilno,ad, soyad ve maaş bilgisini yazdırınız.

Çözüm:

Oracle çözümü:

```
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.sicilno ||
'|' || p_personel.ad || '|' ||
        p_personel.soyad || '|' || p_personel.maas);
    end loop;
    close p_kursor;
end;
```

sql server çözümü:

```
declare p_kursor cursor for select sicilno,ad,soyad,maas from personel
declare @sicilno char(10),@ad varchar(20),@soyad varchar(20),@maas money

open p_kursor

fetch next from p_kursor into @sicilno,@ad,@soyad,@maas

while @@fetch_status=0
begin
    print @sicilno+' '+@ad+' '+@soyad+' '+str(@maas)
    fetch next from p_kursor into @sicilno,@ad,@soyad,@maas
end
close p_kursor
deallocate p_kursor
```

KULLANICI TANIMLI PAKETLER

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

Örneğin yazdırma amacıyla kullandığımız dbms_output bir pakettir ve bu paket içinde çeşitli prosedürler bulunur. Örneğin bu paketteki put_line prosedürü bir satırlık yazı yazar.

Paketler, kullanıcılar tarafından da oluşturulabilir. Paketler ile nesne tabanlı programlama yapılabilir. Yani veri tabanına nesne yönelimli özellik verilebilir. Bu yapı sadece oracle'da vardır.

Paket içinde prosedür, fonksiyon, değişken, kursor gibi veritabanı nesnelere bulunabilir.

Paketler iki kısımdan oluşur. İlk önce paketin tanımı yapılır. Bu tanımda sadece prosedürün ismi, fonksiyon ismi ve parametreleri ayrıca değişkenler bulunur.

İkinci aşamada paketin gövdesi yazılır. Bu gövdede prosedür ve fonksiyonun içinde çalışan kodlar bulunur.

Paket tanımı aşağıdaki gibi yapılır.

```
create or replace package paketAdi is
    tanımlamalar
end paketAdi;
```

paketin gövde yapısı aşağıdaki gibidir.

```
create or replace package body paketAdi is
    gövde kodları, prosedür gövdeleri
end paketAdi;
```

Örnek:

Paket tanımlaması yapalım

```
create or replace package pkg_Personel is
  procedure toplam_maas;
  procedure personel_maas(p_sicilno char);
  p_maas number(15,2);
  procedure personel_kayit(p_sicilno char);
  p_personel personel%rowtype;
end pkg_Personel;
```

paket gövdesini yazalım.

```
create or replace package body pkg_personel 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 pkg_personel;
```

yukarıdaki paketi aşağıdaki örnekte kullanalım

```
begin
  pkg_personel.toplam_maas;
  dbms_output.put_line(pkg_personel.p_maas);

  pkg_personel.personel_maas('2053');
  dbms_output.put_line(pkg_personel.p_maas);

  pkg_personel.personel_kayit('2053');
  dbms_output.put_line(
    pkg_personel.p_personel.ad || ' ' ||
    pkg_personel.p_personel.soyad || ' ' ||
    pkg_personel.p_personel.maas);
end;
```