ORACLE’DA KÜRSÖRLER

Bunun için bulunan sistem değişkenleri

Aşağıdaki sistem değişkenlerinin değeri true yada false dir.

SQL%ISOPEN : kürsör açıkmı 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.

Genel kürsör kullanım kalıbı

declare

 CURSOR kürsörAdi IS SelectDeyimi;

 kayıtTipiDeğişken kürsörAdi%ROWTYPE;

begin

 OPEN kürsörAdi

 LOOP

 FETCH kürsörAdi INTO kayıtTipiDeğişken;

 EXIT WHEN kursörAdi%NOTFOUND;

 komutlar

 END LOOP;

 CLOSE kürsörAdi;

end;

yukarıdaki kodda bulunan EXIT deyimini açıklayalım.

EXIT deyimi döngüden çıkmayı sağlar iki çeşit kullanımı vardır.

EXIT veya

EXIT WHEN koşul

Koşul doğru değer verdiğinde döngüden çıkar. Üstteki EXIT için IF deyimi kullanmak gerekir.

Örnek:

Kürsörü 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 için çözüm.

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

Kullanıcı Tanımlı Paketler

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

Örneğin, yazdırma amacıyla kullandığımız dbms\_output bir pakettir ve bu paket içinde çeşitli prosedürler bulunur. Örneğin put\_line prosedürü ile bir satırlık yazı yazılır.

Paketler, kullanıcılar tarafından da oluşturulabilir. Paketler ile nesne tabanlı programlama yapılabilir. Yani veritabanına nesne yönelimli özellik verilebilir.

Paket içinde, prosedür, fonksiyon, değişken, kürsör gibi nesneler bulunabilir.

Paketler iki kısımdan oluşur. İlk önce paketin tanımı yapılır. Sonra bu paketin gövdesi oluşturulur.

Paketin tanımı aşağıdaki gibi yapılır.

create or replace package paketAdi is

 tanımlamalar

end paketAdi;

Paketin gövdesi aşağıdaki gibi oluşturulur.

create or replace package body paketAdi is

 gövdedeki komutlar

 prosedür ve fonksiyon gövdeleri burada yazılır.

end paketAdi;

tanımlama kısmında sadece değişken adı, prosedür adı, fonksiyon adı ve parametreleri bulunur.

Örnek:

Örnek paket tanımlaması

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;

tanımlanan paketin gövdesi aşağıdaki gibidir.

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 çalıştıralım.

begin

 -- toplam personel maaşı
 pkg\_personel.toplam\_maas;
 dbms\_output.put\_line(pkg\_personel.p\_maas);

 --2053 nolu personelin maaşı
 pkg\_personel.personel\_maas('2053');
 dbms\_output.put\_line(pkg\_personel.p\_maas);

 --2053 nolu personelin kayıt bilgisi
 pkg\_personel.personel\_kayit('2053');
 dbms\_output.put\_line( pkg\_personel.p\_personel.ad|| ' ' || pkg\_personel.p\_personel.soyad);
end;

Put’lu çıktı

declare
 i number;
begin
for i in 1..5
loop
 dbms\_output.put(i||',');
end loop;
 dbms\_output.put\_line('');

end;

Put\_line’lı çıktı

declare
 i number;
begin
for i in 1..5
loop
 dbms\_output.put\_line(i||',');
end loop;
 dbms\_output.put\_line('');

end;