ORACLE’A GİRİŞ

Oracle ile SQL Server’ı karşılaştıralım,

1. Oracle’da veritabanı yerine kullanıcı oluşturulur. Kullanıcılar veritabanı gibi davranır.
2. Tablo oluşturma, yapısını değiştirme, silme kodları aynı.
3. View oluşturma ve değiştirme aynı
4. Select deyimi yapısı aynı (select, from, where , group by , order by, having gibi aynı şekilde kullanılır)
5. Case ifadesi aynı
6. Operatörler aynı (+, -, \*, /, and, or, not, like, in, between, exists, is null, is not null vs.)
7. Sorgulama dili olarak sql server Transact SQL, Oracle PL/SQL dilini kullanır.
8. Oracle’da tablo yapısı içinde otomatik artan kolon yoktur(yani identity kolon).Otomatik artan kolon yapısı ayrı bir nesne olarak oluşturulur.
9. Sütun ve tablo takma adları aynı şekilde verilir.
10. Stored procedure ve fonksiyon yapıları farklıdır.
11. Blok yapıları farklı
12. Döngü yapıları farklı
13. Kısıtlamalar aynı (not null, primary key, foreign key vs. )
14. Yetki verme, yetki alma, yasaklama gibi deyimler benzer yapıdadır.
15. Her ikisi de ANSI 92 standardına uygundur.

**Oracle’da kullanılan veri tipleri:**

Oracle ANSI 92 standardında tanımlanan tiplere uyumlu ancak en çok aşağıdaki veri tipleri kullanılır.

CHAR : sql server karşılığı char

VARCHAR2: SQLserverda varchar

NUMBER: sayı, tamsayı, ondalıklı sayı , sql karşılığı tinyint, smallint, int, bigint, Money vs.

Kullanımı number(uzunluk) veya

 Number(uzunluk,ondalık)

 Number(5) number(15,2)

DATE : sql karşılığı DATETIME

Oracle’da en yetkili kullanıcı adı : SYSTEM

Şifresi, kuruluş esnasında verilir.

Örnek:

Bu veritabanı üzerinde KENAN isminde bir kullanıcı oluşturalım. Bu kullanıcıya gerekli yetkileri verelim.

create user KENAN identified by "123456"
default tablespace users
temporary tablespace temp
profile default
quota unlimited on users;

yetki verelim

GRANT connect, aq\_user\_role, create table, create view,create session TO KENAN

Öğrenci tablosu oluşturalım

CREATE TABLE Ogrenci ( OkulNo char(9) PRIMARY KEY NOT NULL,
Adi varchar2(20) NOT NULL,
Soyadi varchar2(20) NOT NULL,
Adresi varchar2(100), ilPlakaNo number(2), Donem char(9),FakulteKodu char(3), BolumKodu char(2))

Aşağıdaki verileri girelim

insert into ogrenci(okulno,adi,soyadi,adresi,ilplakano,donem,fakultekodu,bolumkodu)
values ('06060','Ahmet','UZUN', 'beykent',34,'2005-2006','FEF','BP');

insert into Ogrenci (OkulNo, Adi, Soyadi, Adresi, ilPlakaNo, Donem, FakulteKodu, BolumKodu)
VALUES ('07019','Ali','YILMAZ',
'İstanbul cad',35,
'2005-2006','MUH','BM');

insert into Ogrenci (OkulNo, Adi, Soyadi, Adresi, ilPlakaNo, Donem, FakulteKodu, BolumKodu)
VALUES ('07022','Ayşe','YILMAZ','beykent',34,
'2006-2007','FEF','AB');

insert into Ogrenci (OkulNo, Adi, Soyadi, Adresi, ilPlakaNo, Donem, FakulteKodu, BolumKodu)
VALUES ('06011','Yaren','AKPINAR',NULL,22,
'2005-2006','MUH','AC');

insert into Ogrenci (OkulNo, Adi, Soyadi, Adresi, ilPlakaNo, Donem, FakulteKodu, BolumKodu)
VALUES ('07012','Gamze','AKTAŞ',
'Kovboy cad',34,'2005-2006','MUH','AC');

insert into Ogrenci (OkulNo, Adi, Soyadi, Adresi, ilPlakaNo, Donem, FakulteKodu, BolumKodu)
VALUES ('05001','Öznur','AKPINAR',
'Amerikan pasaj',35,
'2004-2005','FEF','BP');

insert into Ogrenci (OkulNo, Adi, Soyadi, Adresi, ilPlakaNo, Donem, FakulteKodu, BolumKodu)
VALUES ('06013','Asal','ALTAY',NULL,28,
'2006-2007','MUH','BM');
commit;

**Sorgulama yapalım.**

OkulNo değeri 06060 olan öğrencinin adı ve soyadını veren sorguyu yazınız.

SELECT o.adi, o.soyadi FROM Ogrenci o where o.okulno='06060'

Örnek:

BP,AB,CA bölümlerinde okuyan öğrencilerin okulno, adı, soyadı, bölüm kodlarını listeleyen sorguyu yazınız.

SELECT o.okulno, o.adi, o.soyadi, o.bolumkodu FROM Ogrenci o
WHERE o.bolumkodu in ('BP','AB','CA');

Oracle’da iki sütun || sembolü ile birleştirilebilir.

Az önceki kodu aşağıdaki gibi yapalım.

SELECT o.okulno,o.adi||' '||o.soyadi as AdiSoyadi, o.bolumkodu FROM Ogrenci o

WHERE o.bolumkodu in ('BP','AB','CA');

**Oracle’da otomatik artan sütun nasıl tanımlanır?**

Oracle’da tablo tanımı içinde otomatik artan sayı yoktur. Bunun yerine ayrı bir nesne vardır. Bu nesnenin adı SEQUENCE dir.

CREATE SEQUENCE isim

INCREMENT BY ArtımDeğeri --varsayılan 1

START WITH BaşlangıçDeğeri --varsayılan 1

MAXVALUE MaksimumDeğer

MINVALUE MinimunDeğer;

Örnek:

CREATE SEQUENCE sq\_test

INCREMENT BY 5

START WITH 100;

100 den başla 5’şer 5’şer say

Çalıştırdık. SQ\_TEST adında oluştu.

Bunu select deyimi ile kullanabiliriz.

SELECT sq\_test.nextval from dual;

**Dual tablosu:** tek bir sütunu ve kayıdı bulunan tablodur. Sistem değişkenleri ve ifadeleri hesaplamak için kullanılır. Veritabanındaki tüm kullanıcılara açıktır.

**sqAdi.nextval**: kullanımı bir sonraki numarayı getirir. Her çağrıldığında numara artar.

**sqAdi.currval** : sequencenin şu anki değerini verir.

Örnek:

*Bir test tablosu oluşturalım:*

create table test (sayi number(5), ADI VARCHAR2(20));

*Bu tabloya veri girelim*

insert into test(sayi,adi) values (sq\_test.nextval, 'kenan');

insert into test(sayi,adi) values (sq\_test.nextval, 'Ali');

insert into test(sayi, adi) values (35,'Ayşe');

insert into test(sayi,adi) values (36,'Fadime');

commit;

create user HAFTA13B2 identified by "1234"
default tablespace users
temporary tablespace temp
profile default
quota unlimited on users;

grant all privileges to HAFTA13B2;

SELECT …INTO …. Deyimi

Sorgudan elde edilen değerin değişkenlere aktarılmasını sağlar. Sorgudan tek satır dönmesi gerekir. Çok satır dönerse hata mesajı verir.

SQL server’da aşağıdaki gibi yapılırdı.

**SELECT @ortamalama= avg(notu) from notlar**

Oracle için genel ifade;

SELECT kolonİsimleri INTO değişkenİsimleri FROM TabloAdi WHERE koşul

şeklindedir.

Örnek:

2053 nolu personelin adını ve soyadını değişkenlere aktarınız ve bu değişkenleri yazdırınız.

Çözüm:

declare
 p\_ad varchar2(20);
 p\_soyad varchar2(20);
begin
 **select** ad, soyad **into** p\_ad, p\_soyad from personel where sicilno='2053';

 dbms\_output.put\_line(p\_ad || ' ' || p\_soyad);
end;

yukarıdaki kodu açıklayalım

declare deyimi : değişken tanım bloğudur.

begin…end : çalışacak kodlar yazılır.

p\_adi, p\_soyadi : değişkenlerdir.

select ad,soyad into p\_ad, p\_soyad ….

ile personel tablosundaki ad kolonu değeri p\_adi değişkenine, soyad kolonu değeri p\_soyad değişkenine aktarılır.

dbms\_output : çıktıyı sağlayan oracle paketinin adıdır. Put\_line bu paketin içindeki, yazdırmayı sağlayan prosedürdür.

PL/SQL de atama işlemi:

Atama yapmak için := sembolü kullanılır.

c := a + b;

Örnek :

2053 nolu personelin adını ve soyadını isim değişkenine aktarınız.

declare
 p\_ad varchar2(20);
 p\_soyad varchar2(20);
 p\_isim varchar2(20);
begin
 select ad, soyad into p\_ad, p\_soyad from personel where sicilno='2053';

 p\_isim := p\_ad ||' '||p\_soyad;
 dbms\_output.put\_line(p\_isim);
end;

atama değişken tanımlanırken de yapılabilir.

Örnek

declare

 para number(15,2):=2500;

yukarıda para isimli değişkene değer 2500 atanmıştır.

**PL/SQL Blokları**

* Normal blok
* Fonksiyon bloğu
* Prosedür bloğu

Normal Blok

Az önceki uygulamalar, normal blok yapısına uygun uygulamalardır.

Yapısı:

declare

 değişken tanımlar

begin

 komutlar

EXCEPTION -- hata yakalamak istendiğinde

 Hata durumunda çalışacak kodlar

end;



Örnek:

10 nolu departmanda çalışan personele %15 zam yapan kodu yazınız. Zam oranı değişkeni tanımlayınız ve bu değişkene zam oranını atayınız.

declare
 zam\_orani number := 0.15;
begin
 update personel set maas=maas\*(1+zam\_orani) where departman=10;
 commit;
end;



Fonksiyon Bloğu

Oracle üzerinde fonksiyon oluşturur. Bu fonksiyonlar scaler değerli fonksiyonlardır. Oracle fonksiyonları tablo döndüremez.

Create or replace function fonksiyonAdi

(parametreler)

return dönüşTipi

IS

Değişken tanımları

Begin

 Komutlar

EXCEPTION

 Hata durumunda çalışacak kodlar

end;

Örnek:

Sicil numarası verilen personelin maaşını bulan fonksiyon oluşturunuz. Bu fonksiyondan 2053 nolu personelin maaşını bulunuz.

Çözüm:

**create or replace function maas\_bul**(**p\_sicilno char**)
**return number**is
 p\_maas number;
begin
 select maas into p\_maas from personel where sicilno=p\_sicilno;
 return p\_maas;
end;

şimdi bu fonksiyondan yararlanarak 2053 nolu personelin maaşını bulalım.

declare
 maas number;
begin
 maas := maas\_bul('2053');

 dbms\_output.put\_line(maas);
end;

**create** **or** **replace** **function** maas\_bul(p\_sicilno **char**)
**return** **varchar2**
**is**
 p\_maas **number**;
 p\_ad **varchar2**(20);
 p\_soyad **varchar2**(20);
**begin**
 **select** maas,ad,soyad **into** p\_maas,p\_ad,p\_soyad **from** personel **where** sicilno=p\_sicilno;
 **return** p\_maas||' '||p\_ad||' '||p\_soyad;
**end**;

**declare**
 maas\_isim **varchar2**(30);
**begin**
 maas\_isim := maas\_bul('2053');

 dbms\_output.put\_line(maas\_isim);
**end**;

Örnek :

Departman numarası verilen departmanın Maas toplamını veren fonksiyon oluşturunuz.

create or replace function maas\_toplam(p\_departman number)
return number
is
 p\_maas\_toplam number;
begin
 select sum(maas) into p\_maas\_toplam
 from personel where departman = p\_departman;
 return p\_maas\_toplam;
end;

yukarıdaki fonksiyonu kullanalım





Prosedür bloğu

Oracle üzerinde prosedür oluşur.

Yapısı:

create or replace procedure prosedürAdi (parametreler)

IS

 Değişken tanımları

begin

 komutlar

exception

 hata durumunda çalışacak kodlar

end;

örnek

personel tablosuna veri giren bir prosedür oluşturunuz.

Çözüm

create or replace procedure personel\_veri\_gir
(p\_sicilno char, p\_ad varchar2, p\_soyad varchar2, p\_maas number)
is
begin
 insert into personel (sicilno,ad,soyad,maas)
 values(p\_sicilno, p\_ad, p\_soyad,p\_maas);
 commit;
end;

bu prosedürü kullanalım.

begin
 personel\_veri\_gir('4446','Ali','hasan',2500);
end;

Örnek:

Sicil numarası verilen Personelin adını soyadını geri döndüren bir prosedür oluşturunuz.

Prosedürde geri dönüş parametresi out kelimesi ile sağlanır.

Çözüm:

create or replace procedure personel\_isim\_bul
(p\_sicilno char, p\_adi out varchar2,p\_soyadi out varchar2)
is
begin
 select ad, soyad into p\_adi,p\_soyadi from personel
 where sicilno=p\_sicilno;

end;

Şimdi bu prosedürü kullanarak 2053 nolu personelin adını soyadını öğrenelim.

declare
 p\_ad varchar2(20);
 p\_soyad varchar2(20);
begin
 personel\_isim\_bul('2053',p\_ad,p\_soyad);
 dbms\_output.put\_line(p\_ad || ' ' || p\_soyad);

 personel\_isim\_bul(p\_sicilno => '2053',p\_adi => p\_ad,
 p\_soyadi => p\_soyad);
 dbms\_output.put\_line(p\_ad || ' ' || p\_soyad);
end;



