

Oracle Database Express Edition

User: SYS

Home > Administration > Manage Database Users > Create Database User

Create Database User

* Username: DMG

* Password: ●●●●●●

* Confirm Password: ●●●●●●

Expire Password: ☐

Account Status: Unlocked

Default Tablespace: USERS

Temporary Tablespace: TEMP

User Privileges

Roles:

☒ CONNECT ☒ RESOURCE ☐ DBA

Direct Grant System Privileges:

☒ CREATE DATABASE LINK ☒ CREATE MATERIALIZED VIEW ☒ CREATE PROCEDURE

☒ CREATE PUBLIC SYNONYM ☒ CREATE ROLE ☒ CREATE SEQUENCE

☒ CREATE SYNONYM ☒ CREATE TABLE ☒ CREATE TRIGGER

☒ CREATE TYPE ☒ CREATE VIEW

[Check All](#) [Uncheck All](#)

Kullanıcı Oluşturma Ekranı-II

ORACLE Database 10g Üzerinde Örnek Veri Tabanı

Kitabın diğer bölümlerinde çok sayıda veri tabanı örnekleri verilmiştir. Oracle Bölümünde de PL/SQL komutlarını ve bloklarını çalıştırıp uygulamak için ilişkisel bir veri tabanı modeli üzerinde çalışacağız. Veritabanımız hakkında bilgi ve diyagram bu bölümde size verilecektir.

Veri Tabanının ismi, **UEFA Şampiyonlar Ligi Maç Yönetimi** olarak karşımıza çıkmaktadır. Bu veritabanı ile sorgulanmak istenenler:

- Şampiyonlar Ligi takımlarının bilgileri.
- Maç yapılan Stat bilgileri.
- Takımlar arasında yapılan maçlar ve sonuç bilgileri.
- Gruplardaki takımlar ve Puan durumu bilgileridir.

Örnek Veritabanının SQL Kodları

DDL (Data Definition Language) Kodları

Gruplar Tablosunun Oluşturulması

```
Create Table Gruplar
( g_name varchar2(5),
  constraint pk_Gruplar primary key (g_name)
);
```

Stadlar Tablosunun Oluşturulması

```
Create Table Stadlar
(Std_Id number(4),
 isim varchar(50),
 kapasite number(4),
 ac_tar date,
 constraint pk_Stadlar primary key (Std_Id)
);
```

Takımlar Tablosunun Oluşturulması

```
Create Table Takimlar
(tkno numeric(10) not null,
 isim varchar2(50),
 kur_tar date,
 gname varchar2(5),
 butce varchar2(30),
 constraint pk_Takimlar primary key (tkno),
 constraint fk_Takimlar foreign key (gname) references Gruplar(g_name));
```

Maçlar Tablosunun Oluşturulması

```
Create Table Maclar
(mac_Id int not null,
 mac_tar date,
 gol_evsahibi smallint,
 gol_konuk smallint,
 tkm_Id_ev numeric(10),
 tkm_Id_konuk numeric(10),
 std_Id number(4),
 constraint pk_Maclar primary key (mac_Id),
 constraint fk_Takim1 foreign key (tkm_Id_ev) references Takimlar(tkno),
 constraint fk_Takim2 foreign key (tkm_Id_konuk) references Takimlar(tkno),
 constraint fk_stad foreign key (std_Id) references Stadlar(Std_Id)
);
```

Not: Yukarıdaki Takımlar tablosunu Create Table komutu ile oluştururken Puan alanını oluşturmadık. Bu alanı yine **ALTER Table** komutu ekleyebiliriz. Kod aşağıdaki gibidir. Bu kodlar kitabın önceki bölümlerinde anlatılmıştır.

Alter Table Takımlar

Add puan Integer;

DML (Data Manipulation Language) Kodları

Stadlar Tablosuna Veri Girilmesi

```
insert into Stadlar values (2345, 'İnönü', 50000, '02/01/1903');  
insert into Stadlar values (234, 'Şükrü Saraçoğlu', 40000, '10/01/1923');  
insert into Stadlar values (23, '24 Kasım', 3000, '12/12/1953');  
insert into Stadlar values (5645, 'Santiago Bernabeu', 55000, '02/01/1900');  
insert into Stadlar values (8014, 'Nou camp', 30000, '01/01/1920');
```

Gruplar Tablosuna Veri Girilmesi

```
insert into Gruplar values ('A');  
insert into Gruplar values ('B');  
insert into Gruplar values ('C');  
insert into Gruplar values ('D');  
insert into Gruplar values ('E');  
insert into Gruplar values ('F');
```

Takımlar Tablosuna Veri Girilmesi

```
insert into Takımlar values(1,'Real Madrid','01/01/1880','A','1000000',0);

insert into Takımlar values(2,'Beşiktaş','21/01/1901','A','1000000',0);

insert into Takımlar values(3,'Cska Moskova','18/01/1900','A','1100000',0);

insert into Takımlar values(14,'Chealsea','01/08/1910','B','800000',0);

insert into Takımlar values(13,'Juventus','11/01/1920','B','10000000',0);

insert into Takımlar values(11,'Dmg United','03/10/1978','B','2000000',0);
```

Maçlar Tablosuna Veri Girilmesi

Takımlar tablosunun verilerinin girdiğimizde puan alanı default (0) olarak girilmiştir. Fakat Maçlar tablosu girildiğinde bu alan ileride güncellenecektir.

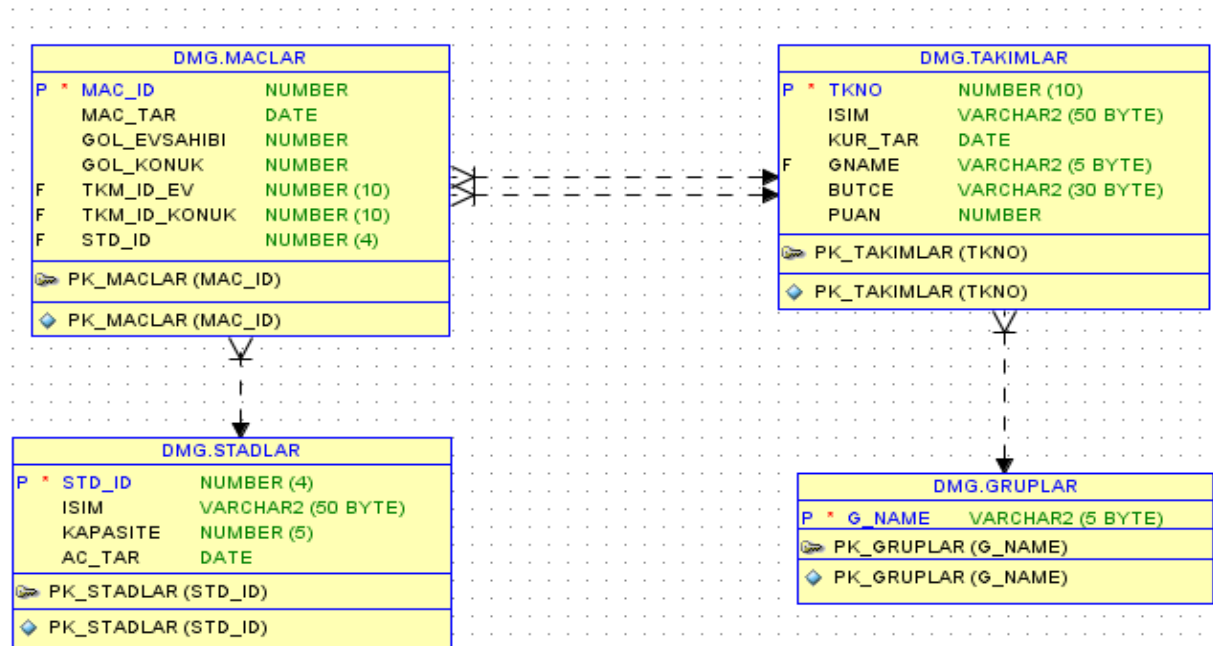
```
insert into Maclar values(23,'12/10/2010',5,3,1,2,5645);

insert into Maclar values(24,'12/12/2010',3,1,1,3,5645);

insert into Maclar values(20,'12/11/2011',1,1,14,13,5645);

insert into Maclar values(19,'19/11/2011',1,0,11,13,5645);
```

Örnek Veritabanının İlişkisel Diyagramı



SQL Yapısal Sorgulama Dili

Kitabın diğer bölümlerinde SQL ile ilgili komutlar işlenmiş ve örnekler verilmiştir. Bu bölümde de Oracle'ın veri sorgulama dili olan PL/SQL'e geçiş yaparken SQL özeti yapmak isterim. SQL, İlişkisel veritabanlarında veriye ulaşmak ve bu veriyi okumak, yorumlamak, değiştirmek ya da verinin depolandığı birimler ile her türlü işlemlerin yapıldığı evrensel veri tabanı kodlarıdır. SQL aslında dört kategoride toplanabilir.

Veri Tanımlama Dili (DDL- Data Definition Language)

Verinin tutulacağı tablo ya da veritabanı oluşturma komutları olarak bilinen komutları içeren SQL kısmıdır. Yapısal değişiklik yaparlar. Veriyle değil, verinin yerleştiği yapıyla uğraşırlar. Tanımlamalar bu kısımda yapılmaktadır. Bunlardan en genel kullanılanlar aşağıdaki komutlarıdır.

Komut	İşlevi
CREATE	Veritabanında nesne yaratır.
ALTER	Veritabanının yapısını değiştirir.
DROP	Veritabanından nesne siler.
TRUNCATE	Tablodaki kayıtları içerdikleri alan ile birlikte siler.
COMMENT	Yorum ekler.
RENAME	Nesnenin adını değiştirir.

Not: Bu komutları kullanmak yerine Veri tabanı yönetim konsolundan bu işlemleri görsel olarak yapmak daha kolaydır. Fakat bir veri tabanı uzmanı olacak kişinin tüm komutlara hâkim olması gerekmektedir.

Veri İşleme Dili (DML - Data Manipulation Language)

Verileri tabloya ekleme, tablodaki verileri silme ya da veriler üzerinde yapılan değişiklikler için kullanılan komutlarıdır. Bu komutlar program kodları içerisinde yada yazılan programlarda çağırılan prosedür ve fonksiyonların içinde çok kullanılmaktadır. Örnek olarak bir web sitesine kayıta girilen bilgiler veri tabanına **insert** komutu ile girilmektedir.

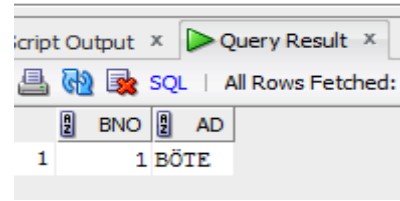
Komut	İşlevi
SELECT	Veritabanından kayıtları getirir.
INSERT	Tabloya kayıt ekler.
UPDATE	Tablodaki kayıtları günceller.
DELETE	Tablodan sadece kayıtları siler.
MERGE	UPSERT işlemi (ekleme veya güncelleme)
CALL	PL/SQL veya Java alt programı çalıştırır.

Not: Burada MERGE komutu ile ilgili bir örnek vermek isterim. Çünkü bu komut Oracle’da aktif olarak kullanılmaktadır. Amacı bir tabloya veri girilirken veri varsa güncelleme yoksa kayıt işlemini tek SQL sorgusu ile yapabiliriz.

Örnek: Bölüm tablosunda 1 numaralı kayıt var ise bunu güncelleyen yok ise BÖTE isimli bir kayıt girmesini sağlayan kodu yazınız.

```
merge into bolum b
using dual s on (b.bno=1)
when matched then
update set ad='BÖTE'
when not matched then
insert(bno,ad) values (1,'BÖTE');
```

```
select * from bolum;
```



BNO	AD
1	1 BÖTE

İşlem Kontrol Dili (TCL-Transaction Control Language)

Transaction veritabanındaki işlem ya da işlem bloğudur. Bu işlemlerin tablo ya da şema gibi depo birimleri ya da veriler üzerinde yapacaklar değişikliklerin geçerli olmasını ya da geri almayı sağlarız. Böylece transactionları kontrol etmiş oluruz. Buna örnek bir para transferi işlemidir. Para transferi yapılmadan **transaction** başlatılır. Bir sorun çıkmazsa sonunda **commit** ile bitirilir. Sorun çıkarsa işlem **rollback** ile geri çekilir.

Komut	İşlevi
COMMIT	Yapılan işlemleri onaylar.
SAVEPOINT	Daha sonra geri dönülecek bir dönüş noktası belirler.
ROLLBACK	Son COMMIT’e kadar olan kısmı geri alır.