

Veritabanı uygulama dersi - 1

DDL	DML
Tablo yaratma CREATE TABLE tablo_adi (sütun_listesi)	Satır Ekleme INSERT INTO tablo [(sütun_listesi)] VALUES (değer_listesi)
Sütun tanımı: sütun_adi veri tipi	Satır Silme DELETE FROM tablo [WHERE koşul]
Tablo silme DROP TABLE tablo_adi	Satır Güncelleme UPDATE tablo SET atama_listesi [WHERE koşul]
Tablo adı değiştirme RENAME tablo_adi TO yeni_ad	
Sütun ekleme ALTER TABLE tablo_adi ADD sütun_tanımı	

DDL (Data Definition Language) Komutları

Yeni tablo yaratmak için **CREATE TABLE** komutu kullanılır. Kullanım şekli:

```
Create table TableName  
(  
FieldName FieldType (Width) PrimaryKey,  
FieldName FieldType (Width) Default (Default value),  
FieldName FieldType (Width) not null,  
FieldName FieldType (Width),  
FieldName FieldType,  
Primary key (PkeyField1,PkeyField2)  
);
```

(TableName) : yaratılmak istenen tablo adı.

(FieldName) : kolon/sütun adı.

(FieldType) : Kolon/sütun tipi . Char(n), Varchar(n), Numeric(p,s), double(p,s) , Int4, Integer, bigint, int8, real, float4, double, float8, Smallint, byte, boolean tiplerinden biri olabilir. Daha fazla bilgi için Bkz. : <http://www.postgresql.org/docs/8.4/interactive/datatype.html>

(Width) : Alan genişliği.

(Default Value) : Default ifadesi ile beraber kullanılır. Eğer kayıt girilirken bu alana veri girilmemişse default value ne ise o değer kullanılır.

PrimaryKey : eğer bir kolon primary key olarak tanımlanacaksa sonuna primarykey yazılır.

Primary Key () : Bu ifade eğer bir veya birden fazla kolon seçilecekse kolon adı yanına değil ayrı bir satır olarak yazılır. primary key komutunun yanında parantez içine alan adları yazılır.

Örnek 1 :

ADI	SOYADI	NO
char[20]	char[20]	numeric

Öğrenci tablosu ADI, SOYADI, NO olmak üzere 3 hücreden oluşan bir tablo olsun. SQL ile bu tabloyu yaratalım:

Create table Ogrenci
(
Adi Char(20) not null,
Soyadi Char(20) not null,
No Numeric,
Primary Key (**No**)
);

- Var olan bir tabloya yeni kolon eklemek için **ALTER TABLE ... ADD** kalıbı kullanılır:

Örnek 2 : Öğrenci tablosuna char(11) tipinde “Telefon” isimli bir kolon/sütun ekleyelim:

Alter table Ogrenci **add** Telefon char(11);

- Yaratılan tabloyu ortadan kaldırmak için **DROP TABLE** komutu kullanılır:

Örnek 3 : Yarattığımız öğrenci tablosunu ortadan kaldıralım:

Drop table Ogrenci;

DML (Data Manipulation Language) Komutları

Veritabanında oluşturduğumuz tablolara veri girişi yapmak; mevcut verileri değiştirmek veya verileri silmek istediğimizde DML komutlarını kullanırız.

Tabloya yeni satır eklemek için **INSERT** kullanılır:

Insert Into Tablo1 (Alan1,Alan2) **Values** ('String İfade',Sayısal İfade ...)

Tablodan belli satırları silmek için **DELETE FROM** kullanılır: (Tablodaki bilgilerin tamamı silenecekse **Where** kullanılmaz!)

Delete From Tablo1 **Where** Silme şartı

Belirtilen kriterlere göre tablodan ilgili kayıt/kayıtları alarak değerlerini değiştirmeye yarayan SQL komutu **UPDATE ... SET** kalıbıdır: (Tablodaki satırların tümü aynı değişime uğrayacaksa **Where** kullanılmaz!)

Update Tablo1 **set** Alan1='String İfade', Alan2=Sayısal İfade, ...

Where Güncelleme Şartı

DML kullanımına örnekler:

ADI	SOYADI	NO
metin[20]	metin[20]	sayı

Örnek 1 : Öğrenci tablosu ADI, SOYADI, NO olmak üzere 3 hücreden oluşan bir tablo olsun. Bu tabloya AD = "Serkan" SOYAD = "Türkel" NO = 4683 bilgilerini ekleyen SQL cümlesini yazın:

Insert Into ogrenci (AD,SOYAD,NO) **Values**('Serkan','Türkel',4683)

Örnek 2 : Yine öğrenci tablosundan numarası 4556 olan öğrencinin kaydını silen SQL cümlesiniz yazınız:

Delete From öğrenci **Where** öğrenci.NO=4556

Örnek 3 : Öğrenci tablosuna eklediğimiz 4683 numaralı Serkan Türkel 'in adını ve soyadını, Adı = "Ali", Soyadı = "Sert" şeklinde değiştiren SQL cümlesini yazınız:

UPDATE öğrenci **SET** AD = 'Ali', SOYAD = 'Sert'
Where öğrenci.NO = 4683

Sorgulamalar

Veritabanında bulunan tablolardaki bilgileri sorgulamak için de yine SQL dilini kullanırız.

Tablolardaki belli bilgileri listelemek için **SELECT** komutu kullanılır. Kullanımı:

Select TabloAdı.Alan1, TabloAdı.Alan2, ... [*] **From** TabloAdı **Where** Sorgulama şartı

Sorgulama Örnekleri:

Örnek 1 : "Öğrenci" tablosundan ADI ve SOYADI hücrelerini seçen bir sorgu oluşturun.

Select öğrenci.ADI, öğrenci.SOYADI **From** öğrenci

Eğer tüm hücreleri seçin deseydik; o zaman yazacağımız SQL cümlesi şöyle olmalı idi:

Select öğrenci.ADI, öğrenci.SOYADI, öğrenci.NO **From** öğrenci
yada
Select * **From** öğrenci

Örnek 2 : Yine öğrenci tablosunda bulunan kayıtlardan Adı Serkan olan öğrencileri seçmemizi sağlayan SQL cümlesi şöyledir:

Select * **From** öğrenci **Where** öğrenci.ADI = "Serkan"

Örnek 3 : Yada soyadında " r " harfi geçen öğrencileri aşağıdaki SQL cümlesi ile seçeriz:

Select * **From** öğrenci **Where** öğrenci.SOYADI **like** "%r%"

Örnek 4 : Eğer Tabloda bulunan kayıtlardan diyelim ki; numarası **1044 ile 2866** arasında olan öğrencileri seçmek istersek aşağıdaki SQL cümlesini kullanırız:

Select * **From** öğrenci **Where** 1044<öğrenci.NO<2866