

## SQL GİRİŞ

Veritabanı işlemleri yaparken, bilgilerin ortak özelliklerine göre sorgularının yapılmasını SQL dili ile sağlarız. SQL dili ile bir Query yani sorgulama oluşturmak oldukça basittir. SQL dilinde kullanılan belli başlı önemli komutlar aşağıda anlatılmaktadır. Bundan önce Select Komutu ile SQL de kullanılan komutları kalıp halinde yazalım, daha sonra bunları açıklayacağız.

**Select** Tablo1.Alan1, Tablo1.Alan2, ... [ \* ] **From** TabloAdı

**Where** Sorgulama İfadesi

**Group By** Gruplanacak Hücreler

**Order By** Küçükten büyüğe yada Büyükten küçüğe Hücrelerin sıralanması

**Having** Gruplanan yada birleştirilen kayıtların hangi kritere göre sorgulanması

Select ile kullanılmayan SQL komutlarından bazılarıda şunlardır :

**Insert Into** Tablo1 (Alan1,Alan2)

**Values** ('String İfade',Sayısal İfade ...)

**Delete \* From** Tablo1 **Where** Sorgulama İfadesi

**Update** Tablo1

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

**Where** Sorgulama İfadesi

### Tarih Biçimleri:

Yerel SQL tarih tanımlarını Amerikan tarih biçimine göre ayarlar. Diğer tarih biçimlerini desteklemez. Amerikan tarih biçimi AA/GG/YY yada AA/GG/YYYY olarak iki çeşittir. Tarih kullanımında bir hataya mahal vermemek için tarih değerleri tek yada çift tırnak içine alınırlar. Eğer tarih değerini tırnak içine almazsak örneğin 17/7/2000 tarihini 17 bölü 7 bölü 2000 olarak algılayacaktır. Aşağıda tarih değerinin SQL ifadelerinde nasıl kullanılacağına dair bir örnek verilmiştir.

```
SELECT *  
FROM orders  
WHERE (saledate <= "1/23/1998")
```

### SQL' in Bölümleri :

SQL, temel olarak aşağıdaki bölümlerden oluşmaktadır.

- **SELECT:** Seçilecek alanların yazıldığı bölümdür.
- **FROM:** Kullanılacak tabloların yazıldığı bölümdür.
- **WHERE:** Sorgu şartlarının yazıldığı bölümdür.
- **GROUP BY:** Belli alanlara göre kayıtları gruplayıp getirmek için kullanılan bölümdür.
- **HAVING:** Group By' in kullanıldığı SQL'lerdeki grupla ilgili şart bölümüdür.
- **ORDER BY:** Sorgu sonucunda dönen kayıtlar için sıralamanın tanımlandığı bölümdür.

## SQL KOMUTLARI

## FROM Deyimi

From deyimi sorgulanacak ve kullanılacak bilgilerin hangi tablodan alınacağını belirtir. Tablodan hangi hücrelerin içindeki bilgilerin kullanılacağını ise; TabloAdı.Hücre1, TabloAdı.Hücre2, ... veya \* jokeri ile belirleriz. Bir örnek ile daha iyi anlayacağız.

Öğrenci Tablosu;

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

**Örnek :** Öğrenci tablosu ADI, SOYADI, NO olmak üzere 3 hücreden oluşan bir tablo olsun. SQL ile, Bu tablodan ADI ve SOYADI hücrelerini seçerek yeni bir tablo (Bu bir sorgudur{Query} ) oluşturunuz.

**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

## WHERE Deyimi

Where deyimini de ; bir tablodan istediğimiz hücreleri seçerken, o tabloda bulunan kayıtlardan hangilerini, hangi kriterlere göre almak istersek kullanırız. Yani tabloda bulunan kayıtlardan hangilerini almak istiyorsak istediğimiz koşulu where deyiminden sonra kullanarak bu kayıtları elde edebiliriz. Aşağıdaki örnekleri inceleyelim...

**Örnek 1 :** Yine öğrenci tablosunda bulunan kayıtlardan Adı Serkan olan öğrencileri seçmemizi sağlayan SQL cümlesi şöyledir..:

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

**Örnek 2:** Yada soyadında " r " harfi geçen öğrencileri aşağıdaki SQL cümlesi ile seçeriz..:

**Select \* From** öğrenci **Where** öğrenci.SOYADI = "%r%"

**Örnek 3:** 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

#####

## GROUP BY Deyimi

Group by deyimi **SUM, COUNT** kullanarak toplam bir sonuç ile bir tablodan istenilen kritere göre istenilen hücreler alınır ve yine group by 'dan sonra yazılan hücrelere göre gruplanır. GROUP BY isteğe bağlıdır. SELECT deyiminde SQL toplam işlevi yoksa özet değerler gözardı edilir.

**Örnek 1 :** Yine öğrenci tablomuzu kullanarakdan soyadı Türkel olan öğrencileri seçerek AD, SOYAD ve NO fieldlarına göre gruplandıralım...:

**Select** öğrenci.AD, öğrenci.SOYAD, öğrenci.NO **From** öğrenci **Where** öğrenci.SOYAD='Türkel' **Group By** öğrenci.AD, öğrenci.SOYAD, öğrenci.NO

#####

## **ORDER BY Deyimi**

Order By deyimi ile de; sorgulama sonucunda bulunan kayıtlar verilen hücrelere göre, Büyükten küçüğe yada Küçükden büyüğe doğru sıralanır.

**Örnek 1 :** Adı Serkan olan Öğrencilerin numaralarını küçükden büyüğe doğru sıralayınız..:

**Select \* From** öğrenci **Where** öğrenci.AD='Serkan' **Order By** öğrenci.NO [asc]

asc yi yazmasakda burada default değer olduğu için küçükden büyüğe doğru sıralama yapardı. Eğer büyükden küçüğe doğru sırala dese idik o zaman şöyle bir SQL cümlesi yazmalıydık...:

**Select \* From** öğrenci **Where** öğrenci.AD='Serkan' **Order By** öğrenci.NO Desc

#####

## **HAVING Deyimi**

Having Deyimi de; GROUP BY yan tümcesi olan bir SELECT deyiminde hangi gruplandırılmış kayıtların görüntüleneceğini belirler. Yani GROUP BY kayıtları birleştirdikten sonra, HAVING deyimide, HAVING yan tümcesinin koşullarını sağlayan ve GROUP BY yan tümcesi ile gruplandırılmış kayıtları görüntüler.

**Örnek 1 :** öğrenci tablosundan AD, SOYAD ve NO fieldlarını alıp bunları gruplayan ve sonra bunlarıda Numarası 1000 ile 2000 arasında olan kayıtlara göre listeleyen SQL cümlesini yazınız..:

**Select** öğrenci.AD, öğrenci.SOYAD, öğrenci.NO **From** öğrenci  
**Group By** öğrenci.AD, öğrenci.SOYAD, öğrenci.NO  
**Having** 1000<öğrenci.NO<2000

#####

## **INSERT INTO Deyimi**

Insert Into Deyimi bir tabloya bir veya daha çok sayıda kayıt eklemeye yarayan SQL komutudur. Buna ekleme sorgusu da denir. Esas kullanılma kalıbı aşağıdaki gibidir...:

### **Çok sayıda kayıt ekleme sorgusu:**

INSERT INTO TabloAdı [(alan1[, alan2[, ...]])] [IN dışveritabanı]  
SELECT [kaynak.]alan1[, alan2[, ...]] FROM tabloifadesi

### Tek kayıt ekleme sorgusu:

INSERT INTO TabloAdı [(alan1[, alan2[, ...]])] VALUES (değer1[, değer2[, ...]])

**Örnek 1 :** öğrenci tablosuna AD = "Serkan" SOYAD = "Türkel" NO = 4683 bilgilerini ekleyen SQL cümlesini yazın..:

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

#####

### DELETE Deyimi

Delete deyimini bir tablodan bir yada daha fazla kayıt silmek için kullanırız. aşağıdaki gibi bir yazılış kalıbı vardır..:

**DELETE** [Tablo.\*] **FROM** Tablo **WHERE** sorgulama ifadesi

**Örnek 1 :** Yine öğrenci tablosundan numarası 4556 olan öğrencinin kaydını silen SQL cümlesiniz yazınız..:

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

#####

### UPDATE Deyimi

Belirtilen kriterlere göre tablodan ilgili kayıt/kayıtları alarak değerlerini değiştirmeye yarayan SQL komutudur. Kullanılış Kalıbı aşağıdaki gibidir..:

**UPDATE** Tablo **SET** yenideğer **WHERE** Sorgulama İfadesi

**Örnek 1 :** öğrenci tablosuna eklediğimiz 4683 numaraları Serkan Türkel 'in adını ve soyadını değiştirelim. Adı = "Ali", Soyadı = "Sert" olsun. SQL cümlesini yazın...:

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

### SQL FONKSİYONLARI

SQL yazılımında sorgulama yapılırken kullanılabilecek aritmetik fonksiyonlar tabloda verilmiştir...

<b>SUM</b>	Yapılan sorgulamada kriteri karşılayan sayısal bilgilerin toplamını verir.
<b>AVG</b>	Yapılan sorgulamada kritere karşılık gelen sayısal bilgilerin aritmetik ortalamasını alır.
<b>COUNT</b>	Yapılan sorgulamada kritere uyan sayısal bilgilerin adetini hesaplar.

<b>MAX</b>	Yapılan sorgulamada kritere uyan sayısal bilgilerin en büyük değerini yakalar.
<b>MIN</b>	Yapılan sorgulamada kritere uyan sayısal bilgilerin en küçük değerini yakalar.
<b>STDEV</b>	Yapılan sorgulamada kritere uyan sayısal bilgilerin standart sapmasını alır.
<b>STDEVP</b>	Yapılan sorgulamada kritere uyan sayısal bilgilerin istatistiksel standart sapmasını alır.
<b>VAR</b>	Yapılan sorgulamada kritere uyan sayısal bilgilerin varyansını alır.
<b>VARP</b>	Yapılan sorgulamada kritere uyan sayısal bilgilerin istatistiksel varyansını alır.