

SQL NOTLARI

1. SELECT: Tablolardan belirli verileri sorgulamak için kullanılan temel SQL komutu SELECT'tir. Bu komut, veri tabanından belirli sütunlardaki verileri seçmenizi sağlar. İşte SELECT komutunun temel kullanımı:

```
Select sütun1, sütun2, ... from tablo  
Select * from Person.Person  
Select FirstName from Person.Person  
Select Top 10* from Person.Person  
Select Top 20 Suffix from Person.Person  
Select Top 10 Percent* from Person.Person
```

- sütun1, sütun2, ...: Seçmek istediğiniz sütunları belirtir.
 - tablo: Verilerin bulunduğu tabloyu belirtir.
2. INSERT: Bu komut bir veri tabanına yeni kayıt eklemek için kullanılır. Temel kullanımı şu şekildedir:

```
Insert Into tablo (sütun1, sütun2, ...)  
Values (değer1, değer2, ...)  
Insert Into calisan(calisanId,isim,soyisim,cinsiyet)  
Values('3','Emre','Akin','E')
```

- tablo: Veri eklemek istediğiniz tabloyu belirtir.
- (sütun1, sütun2, ...): Eklemek istediğiniz sütunları belirtir.
- VALUES (değer1, değer2, ...): Eklemek istediğiniz değerleri sütun sırasına göre belirtir.

3. UPDATE: Bir veri tabanında güncelleme yapmak için kullanılır. Temel kullanımı şu şekildedir:

```
UPDATE tablo  
SET sütun1 = değer1, sütun2 = değer2, ...  
WHERE koşul;  
Update calisan set soyisim = 'Kaya' where calisanId = 1
```

- tablo: Güncellenmek istenen tabloyu belirtir.
- SET sütun1 = değer1, sütun2 = değer2, ...: Güncellenecek sütunları ve yeni değerlerini belirtir.
- WHERE koşul: Güncelleme işleminin hangi koşula uyan satırları etkileyeceğini belirtir. Bu kısım opsiyoneldir, ancak genellikle kullanılır.

4. DELETE: Bir veri tabanındaki mevcut kayıtları silmek için kullanılır. Temel kullanımı şu şekildedir:

```
DELETE FROM tablo WHERE koşul;  
Delete from calisan where calisanId = 8
```

- tablo: Kayıtların silineceği tabloyu belirtir.
- WHERE koşul: Hangi koşula uyan kayıtların silineceğini belirtir. Bu kısım opsiyoneldir, ancak genellikle kullanılır.

5. WHERE: Bir veri tabanında belirli koşullara göre verileri filtrelemek için kullanılır. Temel kullanımı şu şekildedir:

```
DELETE FROM tablo
WHERE koşul;
Select * from calisan where calisanId < 30
Select CalisanIsmi, CalisanDepartman from calisan where calisanId >= 40
```

- tablo: Kayıtların silineceği tabloyu belirtir.
- WHERE koşul: Hangi koşula uyan kayıtların silineceğini belirtir.

6. AND: SQL sorgularında birden fazla koşulu birleştirmek için kullanılan bir mantıksal operatördür. Bu ifade, sorguda belirtilen tüm koşulların bir arada sağlanması durumunda bir kaydın seçilmesini sağlar. İki veya daha fazla koşulu birleştirmek için kullanılır ve her iki koşulun da doğru olması gerekir. Temel kullanımı şu şekildedir:

```
SELECT * FROM tablo WHERE koşul1 AND koşul2
Select * from calisan where calisanId <= 8 and cinsiyet = 'K'
SELECT * FROM müşteriler WHERE şehir = 'İstanbul' AND yaş > 25
```

- SELECT *: Tüm sütunları seç anlamına gelir. Yani, sonuç olarak tüm sütunlara ait veriler getirilecektir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul1 AND koşul2: Bu kısım, sadece belirli bir koşulu sağlayan kayıtları getirmek için kullanılır. AND ifadesi, iki koşulu birleştirir ve her iki koşulun da aynı anda doğru olması gerektiğini belirtir.

7. OR: SQL sorgularında birden fazla koşulu birleştirmek için kullanılan bir mantıksal operatördür. Bu ifade, sorguda belirtilen koşullardan herhangi birinin sağlanması durumunda bir kaydın seçilmesini sağlar. Yani, en az bir koşulun doğru olması, kaydın seçilmesi için yeterlidir. Temel kullanımı şu şekildedir:

```
SELECT * FROM tablo WHERE koşul1 OR koşul2
Select * from calisan where isim = 'Ali' or cinsiyet = 'E'
SELECT * FROM müşteriler WHERE yaş < 25 OR şehir = 'Ankara'
```

- SELECT *: Tüm sütunları seç anlamına gelir. Yani, sonuç olarak tüm sütunlara ait veriler getirilecektir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul1 OR koşul2: Bu kısım, sadece belirli bir koşulu sağlayan kayıtları getirmek için kullanılır. OR ifadesi, iki koşuldan en az birinin doğru olması durumunda kaydın seçilmesini sağlar.

8. IN: SQL sorgularında belirli bir liste içinde bulunan değerlere sahip olan kayıtları seçmek için kullanılır. Bu ifade, bir sütunun belirli değerlere eşit olup olmadığını kontrol etmek için kullanılır. Genellikle, bir sütunun birden çok özel değere sahip olup olmadığını kontrol etmek amacıyla kullanılır. Temel kullanımı şu şekildedir:

```
SELECT * FROM tablo WHERE sütun IN (değer1, değer2, ...)
Select * from calisan where calisanId IN(8,21,16,54)
SELECT * FROM ürünler WHERE kategori IN ('Elektronik', 'Tekstil')
```

- SELECT *: Tüm sütunları seç anlamına gelir. Yani, sonuç olarak tüm sütunlara ait veriler getirilecektir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE sütun IN (değer1, değer2, ...): Bu kısım, belirli bir sütunun içinde bulunan değerlere göre kayıtları seçmek için kullanılır.

9. LIKE: SQL sorgularında belirli bir deseni içeren değerleri seçmek için kullanılır. Genellikle metin sütunlarında kullanılır ve belirli bir karakter dizisi veya deseni içeren kayıtları seçmek amacıyla kullanılır. Temel kullanımı şu şekildedir:

```
SELECT * FROM müşteriler WHERE ad LIKE 'A%'
Select * from calisan where isim like '%a%'
Select * from calisan where isim like 'a%'
Select * from calisan where isim like '%a'
Select * from calisan where isim like '[ab]%'
```

- SELECT *: Tüm sütunları seç anlamına gelir. Yani, sonuç olarak tüm sütunlara ait veriler getirilecektir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE sütun LIKE "": Bu kısım, belirli bir sütunun içinde bulunan değerleri belirli bir desene göre filtrelemek için kullanılır.

LIKE ifadesinde genellikle kullanılan joker karakterleri şunlardır:

- %: Herhangi bir karakter veya karakter dizisini temsil eder.
- _ (alt çizgi): Tek bir karakteri temsil eder.

10. AS(ALIAS): SQL sorgularında sütun veya tablo isimlerine takma ad (alias) vermek için kullanılır. Bu ifade, sorgu sonuçlarını daha okunabilir ve anlaşılır hale getirmek, sütun isimlerini değiştirmek veya hesaplanan sütunların isimlendirilmesi gibi durumlar için kullanılır. Temel kullanımı şu şekildedir:

```
SELECT sütun1 AS takma_ad1, sütun2 AS takma_ad2, ...  
FROM tablo AS takma_ad_tablo  
WHERE koşul  
Select Isim as 'Çalışan İsimleri' from calisan  
Select P.PersonelId as Numara, P.Isim as Adı, P.Soyismi as Soyadı from Personel P  
SELECT Ad AS isim, soyad AS soyisim FROM müşteriler WHERE şehir = 'İstanbul';
```

- SELECT sütun1 AS takma_ad1, sütun2 AS takma_ad2, ...: Sorgu sonuçlarında görünen sütunlara takma ad (alias) verir. Bu, sorgu sonuçlarını daha anlaşılır hale getirmek için kullanılır.
- FROM tablo AS takma_ad_tablo: Sorguda kullanılan tabloya takma ad (alias) verir. Bu, tablo ismini kısaltmak veya okunabilirliği artırmak için kullanılır.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

11. BETWEEN: SQL sorgularında belirli bir aralıktaki değerleri seçmek için kullanılır. Bu ifade, belirli bir değerin minimum ve maksimum değerler arasında olup olmadığını kontrol etmek için kullanılır. Temel kullanımı şu şekildedir:

```
SELECT * FROM tablo WHERE sütun BETWEEN değer1 AND değer2;  
Select * from calisan where calisanId Between 8 and 25  
SELECT * FROM ürünler WHERE fiyat BETWEEN 50 AND 100;
```

- SELECT *: Tüm sütunları seç anlamına gelir. Yani, sonuç olarak tüm sütunlara ait veriler getirilecektir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE sütun BETWEEN değer1 AND değer2: Bu kısım, belirli bir sütunun belirli bir aralıktaki değerlere sahip olan kayıtları seçmek için kullanılır.

12. DISTINCT: SQL sorgularında kullanılan bir anahtar kelime olup, sorgu sonuçlarındaki tekrarlanan değerleri bir kez göstermeyi sağlar. Yani, sorgu sonuçları içinde aynı değeri taşıyan birden fazla kayıt varsa, DISTINCT kullanılarak bu tekrarlanan değerler bir kez gösterilir. Temel kullanımı şu şekildedir:

```
SELECT DISTINCT sütun1, sütun2, ... FROM tablo WHERE koşul;  
Select distinct Isim from calisan
```

- SELECT DISTINCT sütun1, sütun2, ...: Bu kısım, sorgu sonuçlarındaki tekrarlanan değerleri bir kez göstermek için kullanılır. Belirtilen sütunlara göre benzersiz değerleri getirir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

13. ORDER BY: SQL sorgularında sonuç kümesini belirli bir sıra ile düzenlemek için kullanılan bir ifadedir. Bu ifade, sorgu sonuçlarını belirli bir sütuna göre artan (ascending) veya azalan (descending) şekilde sıralamak için kullanılır. Temel kullanımı şu şekildedir:

```
SELECT sütun1, sütun2, ...  
FROM tablo  
WHERE koşul  
ORDER BY sıralama_sütunu [ASC | DESC];
```

```
Select * from calisan order by calisanId  
Select * from calisan order by calisanId asc  
Select * from calisan order by calisanId desc  
Select Top 10 Percent * from Personel order by 2  
Select Top 10 Percent * from Personel order by 1
```

- SELECT sütun1, sütun2, ...: Bu kısım, sorgu sonuçlarında gösterilecek sütunları belirtir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.
- ORDER BY sıralama_sütunu: Sonuçların hangi sütuna göre sıralanacağını belirtir.
- [ASC | DESC]: Opsiyonel olarak, sıralama yöntemini belirtir. "ASC" (artan) varsayılan olarak kullanılır, "DESC" (azalan) ise sıralamanın tersine çevrilmesini sağlar.

14. MAX: SQL sorgularında bir sütundaki en büyük değeri bulmak için kullanılan bir agregat fonksiyonudur. MAX fonksiyonu, belirli bir sütundaki en yüksek değeri döndürür. Temel kullanımı şu şekildedir:

```
SELECT MAX(sütun)  
FROM tablo  
WHERE koşul;  
Select MAX(calisanId) from calisan  
Select MAX(fiyat) from urunler
```

- SELECT MAX(sütun): Bu kısım, belirli bir sütundaki en büyük değeri getirmek için kullanılır.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

15. MIN: SQL sorgularında bir sütundaki en küçük değeri bulmak için kullanılan bir agregat fonksiyonudur. MIN fonksiyonu, belirli bir sütundaki en küçük değeri döndürür. Temel kullanımı şu şekildedir:

```
SELECT MIN(sütun)  
FROM tablo
```

WHERE koşul;

Select MIN(fiyat) from urunler

- SELECT MIN(sütun): Bu kısım, belirli bir sütundaki en küçük değeri getirmek için kullanılır.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

16. SUM: SQL sorgularında belirli bir sütundaki sayısal değerlerin toplamını bulmak için kullanılan bir agregat fonksiyonudur. SUM fonksiyonu, belirli bir sütundaki tüm sayısal değerleri toplar ve sonucu döndürür. Temel kullanımı şu şekildedir:

SELECT SUM(sütun)

FROM tablo

WHERE koşul;

Select SUM(fiyat) from urunler

- SELECT SUM(sütun): Bu kısım, belirli bir sütundaki sayısal değerlerin toplamını getirmek için kullanılır.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

17. AVG: SQL sorgularında belirli bir sütundaki sayısal değerlerin ortalamasını bulmak için kullanılan bir agregat fonksiyonudur. AVG fonksiyonu, belirli bir sütundaki tüm sayısal değerlerin toplamını alır ve bu toplamı sütundaki değerlerin sayısına böler, böylece ortalama değeri elde eder. Temel kullanımı şu şekildedir:

SELECT AVG(sütun)

FROM tablo

WHERE koşul;

Select AVG(fiyat) from urunler

Select AVG(notlar) from ogrenciler

- SELECT AVG (sütun): Bu kısım, belirli bir sütundaki sayısal değerlerin ortalamasını getirmek için kullanılır.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

18. COUNT: SQL sorgularında belirli bir sütundaki veya tablodaki satır sayısını bulmak için kullanılan bir agregat fonksiyonudur. Temel kullanımı şu şekildedir:

SELECT COUNT(sütun)

FROM tablo

WHERE koşul;

Select COUNT(UrunNo) from urun

Select COUNT(*) from urun

- SELECT COUNT (sütun): Bu kısım, belirli bir sütundaki veya tablodaki satır sayısını getirmek için kullanılır. Eğer sütun yerine yıldız (*) kullanılırsa, tüm satır sayısını getirir.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları saymak için kullanılır.

19. UPPER: SQL sorgularında metin sütunlarındaki karakterleri büyük harfe dönüştürmek için kullanılan bir fonksiyondur. Temel kullanımı şu şekildedir:

SELECT UPPER(sütun)

FROM tablo

WHERE koşul;

Select UPPER(isim) from calisan

- SELECT UPPER (sütun): Bu kısım, belirli bir metin sütunundaki karakterleri büyük harfe çevirmek için kullanılır.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

20. LOWER: SQL sorgularında metin sütunlarındaki karakterleri küçük harfe dönüştürmek için kullanılan bir fonksiyondur. Temel kullanımı şu şekildedir:

SELECT LOWER(sütun)

FROM tablo

WHERE koşul;

Select LOWER(isim) from müşteriler

- SELECT LOWER (sütun): Bu kısım, belirli bir metin sütunundaki karakterleri küçük harfe çevirmek için kullanılır.
- FROM tablo: Verilerin alınacağı tabloyu belirtir.
- WHERE koşul: Opsiyonel olarak, belirli bir koşula uyan satırları seçmek için kullanılır.

21. SUBSTRING: SQL içerisinde istediğimiz karakter aralığını çekip almamızı sağlar. Aldığı parametreler sırasıyla; kolon – başlangıç sayısı – bitiş sayısı şeklindedir.

SELECT SUBSTRING('Ahmet Yılmaz', 1, 3) AS ilk_uc_harf

22. NOT NULL: Bu fonksiyon bir alanı her zaman değer içermesi için zorlar. Yani bu alana herhangi bir değer eklemekten yeni bir kayıt eklememize izin verilmez.

```
Create Table calisan(  
calisanId int Not Null,  
Isim varchar(20),  
Soyisim varchar(20),  
Cinsiyet varchar(1))
```

23. UNIQUE CONSTRAINT: Bir sütundaki tüm değerlerin farklı yani benzersiz olmasını sağlar.

```
Create Table calisan(  
calisanId int Not Null unique,  
Isim varchar(20),  
Soyisim varchar(20),  
Cinsiyet varchar(1))
```

NOT: Bir kolonunuzun benzersiz olmasını istiyorsunuz ancak tabloyu oluştururken bu özelliği eklemediniz bunu sonradan aşağıdaki kodlar ile ekleyebiliriz:

```
Alter table calisan add constraint calisanId unique
```

24. CHECK CONSTRAINT: Bir tabloya veri eklerken veya güncellerken belirli bir koşulun sağlanmasını zorunlu kılmak için kullanılan bir kısıttır. CHECK CONSTRAINT bir sütun veya sütun grubu üzerinde tanımlanabilir ve bu sütunlar da belirli bir koşulu kontrol eder. Bu kısıt ihlal edildiğinde, veri ekleme veya güncelleme işlemi başarısız olur ve bir hata döner.

```
Create Table ogrenciler(  
ogrenciId int,  
Isim varchar(20),  
Soyisim varchar(20),  
Yas int,  
CHECK (Yas >= 0 and yas <= 20))
```

NOT: Alter Table ile Check Constraint ekleme;

```
ALTER TABLE tablo_adi ADD CONSTRAINT kisit_adi CHECK (kosul);
```

25. TRUNCATE TABLE: Bir tablo içindeki yer alan verilerin silinmesi için kullanılır. Delete işlemine göre daha hızlıdır ve tabloyu ilk haline döndürür.

```
Truncate table calisan  
Truncate table urun
```


26. TOP: Tablo içerisinde çağırılacak verilerin sayısını belirtmek için kullanılır.

```
Select Top 8 * from Calisan  
Select Top 50 percent * from urun
```

27. DROP: Tabloyu veya veri tabanını silmek için kullanılır. Tablonun veya veri tabanını tamamını silmektedir.

```
Drop table calisan  
Drop table urun  
Drop database hastane
```

28. CREATE: Veri tabanı veya tablo oluşturmak için kullanılan komuttur.

```
Create database okul  
Create table urun
```

29. ALTER: Bir tabloya sütün eklemek, mevcut sütunları silmek veya değişiklik yapmak için kullanılmaktadır. Bir veri tabanı nesnesinin özelliklerini değiştirmek için kullanılır.

```
Alter table calisan add calisan_yas int  
Alter table calisan alter column calisanId smallint
```

30. LEFT: Metinsel verilerin baş kısmından yani sol kısmından başlayarak belirtilen uzunluktaki kısmını döndürmektedir.

```
Select Left (UrunAdi, 4) from urun
```

31. RIGHT: Metinsel verilerin sağ tarafından istenilen kadar değer almasını sağlar.

```
Select Right (UrunAdi, 4) from urun
```

32. LEN: Bir verinin karakter sayısını, uzunluğunu bulmak için kullanılır.

```
Select UrunAdi, LEN (UrunAdi) as 'Harf Sayısı' from urun
```

33. REPLACE: String ifade içindeki bir alanın istenilen başka bir ifade ile değiştirmek için kullanılır.

```
Select UrunAdi, Replace (UrunKategorisi, 'Baklagil', 'Baklaagil') from urun
```

34. REVERSE: Tersten yazdırma fonksiyonudur.

```
Select Reverse (UrunAdi) from urun
```

35. ABS: Mutlak değer elde etmek için kullanılır.

Select ABS (UrunAdedi) as 'Ürün Adedi' from urun

36. CEILING: Verilen sayısal ifadeyi kendisine en yakın ve kendisinden büyük olan tam sayıya yuvarlayan bir fonksiyondur.

Select Ceiling (UrunAdedi) from urun

37. FLOOR: Verilen sayısal ifadeyi kendisine en yakın ve kendisinden en küçük olan tam sayıya yuvarlayan bir fonksiyondur.

Select Floor (UrunAdedi) from urun

38. ROUND: Virgülden sonra kaç basamak gidileceğini belirtir.

Select KitapAdi, KitapFiyati, Round (KitapFiyati, 1) from Kitap

39. POWER: Bir sayının matematiksel olarak üstünü almaya yarayan fonksiyondur.

Select Power (2,5) as 'Üst'
Select Power (5, 8)

40. SQRT: Bir sayının matematiksel olarak karekökünü almak için kullanılır.

Select SQRT (625)
Select SQRT (888)

41. GETDATE: Anlık tarih ve saati verir.

Select Getdate()

42. DATEPART: Verilen tarih ve saat parametrelerini parçalara ayırarak istenilen parçasının alınmasını sağlar.

Select Datepart (Year, Getdate()) as 'Yıl'
Select Datepart (Day, Getdate()) as 'Gün'
Select Datepart (Month, Getdate()) as 'Ay'
Select Datepart (Week, Getdate()) as 'Hafta'

43. DATEDIFF: İki tarih arasında belirtilen zaman aralıklarını belirlemek için kullanılır.

```
Select Datediff (Year, '1923-08-21', '2016-08-11') as 'Yıl Farkı'
```

44. DATEADD: SQL'de bir tarih veya zaman değerine belirli bir zaman aralığı eklemek için kullanılan bir fonksiyondur.

```
SELECT DATEADD (day, 10, '2024-08-30') AS yeni_tarih
```