



08221

Veri Tabanı II

Elbistan Meslek Yüksek Okulu
2015 – 2016 GÜZ Yarıyılı





08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

DISTINCT: Birbirinin aynı olan satırların listelenmemesi için bu ifade kullanılır

```
"select distinct uyeadi from uyeler"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

LIKE: Eger aradigimiz kaydın bulunmasi için tam bir karsilastirma yapamiyorsak

```
"select * from uyeler where uyeadi like '%a'"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

BETWEEN: Kosul belirtirken iki deger arasini belirtmek için kullanilir. Örnek: Yasi 30 ile 40 arasindaki işçilerin kayitlarini listelemek için

```
"select * from uyeler where yas between 30 and 40"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

SUM: Seçilen degerlerin toplamını bulur. işçilerin aldığı toplam ücreti görmek için

```
"select sum(ucret ) from isciler"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

MAX, MIN, AVG: Verilen degerin en büyüğünü, en küçüğünü ve ortalamasını bulur.

MAX en büyük degeri, MIN en küçük degeri, AVG ise seçilen degerlerin ortalmasını bulur.



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

ORDER BY ASC: Tablodan seçtiğimiz kayıtları alfabetik sıralamak için kullanılır.

```
"select * from uyeler order by asc"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

ORDER BY DESC: Tablodan seçtiğimiz kayıtları son kayıt sırasına göre sıralamak için kullanılır.

```
"select * from uyeler order by desc"
```




08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

ALIAS: Genelde tablonun veya kolonun adını başka bir şeymiş gibi değiştirebiliriz.

Uyeler tablosundaki uyeadi kolonunun adını isim yaptık yani isim olarak çağırabiliriz.

```
"SELECT uyeadi AS isim FROM uyeler"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

JOIN: Bazen iki yada daha fazla tablodan veri çekmemiz gerekebilir, bu gibi durumlarda bu methodu kullanırız.

Birbiri ile ilişkilendirilmiş iki tablomuz var biri uyeler diğeri de detaylar olsun.

İki tablodanda aynı anda sorgulama veya veri çekeceğiz. Uyeler tablosunda id=1 ise detaylar

tablosundaki karşılığı ise uyeno=1 olarak yapıyoruz.



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

```
"SELECT * FROM uyeler,detaylar WHERE uyeler.id=detaylar.uyeno"
```

INNER JOIN OLARAK

```
"SELECT * FROM uyeler INNER JOIN detaylar ON  
uyeler.id=detaylar.uyeno"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

LEFT JOIN OLARAK

```
"SELECT * FROM uyeler LEFT JOIN detaylar ON  
uyeler.id=detaylar.uyeno"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

UNION ve UNION ALL : Join methoduna benzer ama sadece aynı kolonlara sahip tabloları

birbirine bağlar. Uyeler ve Detaylar tablosundaki uye adlarını sorgular.

```
"Select uyeadi from uyeler UNION Select uyeadi from detaylar"
```



08221 Veri Tabanı II

T-SQL KOMUTLARI Devamı

Sadece UNION kullanırsak iki tabloa aynı isimler olsa bile tekrarlar.

UNION ALL kullanırsak iki tabloda aynı isimler olsa bile sadece tekini sorgular.

```
"Select uyeadi from uyeler UNION ALL Select uyeadi from detaylar"
```



08221 Veri Tabanı II

T-SQL Fonksiyonlar

Fonksiyon sözdizimi kurulumu;

SELECT fonksiyon(kolon) FROM tablo

sql Temel fonksiyon tipleri:

++ Aggregate (birleşik değerli) fonksiyonlar

++ Scalar (tekil değerli) fonksiyonlar



08221 Veri Tabanı II

T-SQL Fonksiyonlar

Aggregate (birleşik degerli) fonksiyonlar

Bu tip fonksiyonlar birçok deger ile çalışır ama sonucu tek bir degerdir.



08221 Veri Tabanı II

T-SQL Fonksiyonlar

T-SQL uyumlu fonksiyonlar ise ;

AVG(kolon): girilen kolondaki sayıların aritmetik ortalama degerini geri yollar

BINARY_CHECKSUM: tablonun verilen satırındaki ikilik tabandaki checksum degerini yollar

CHECKSUM: tablonun verilen satırındaki checksum degerini yollar



08221 Veri Tabanı II

T-SQL Fonksiyonlar

T-SQL uyumlu fonksiyonlar ise ;

CHECKSUM_AGG: boş degerler haricindeki verilerin checksum degerini yollar

COUNT(kolon): Boş degerler haric, girilen kolondaki satır sayısını yollar

COUNT(*): Verilen tablodaki satır sayısını yollar



08221 Veri Tabanı II

T-SQL Fonksiyonlar

T-SQL uyumlu fonksiyonlar ise ;

COUNT(DISTINCT kolon): Verilen tablodaki satır sayısını yollar, fakat çift veri bulunan satırları bir defa sayar

FIRST(kolon): girilen kolondaki ilk degeri yollar

LAST(kolon): girilen kolondaki son degeri yollar

MAX(kolon): girilen kolondaki en yuksek degeri yollar

MIN(kolon): girilen kolondaki en dusuk degeri yollar



08221 Veri Tabanı II

T-SQL Fonksiyonlar

T-SQL uyumlu fonksiyonlar ise ;

STDEV(kolon): girilen kolondaki basit istatiksel standard sapma degerini yollar

STDEVP(kolon): girilen kolondaki nüfus istatiksel standard sapma degerini yollar

SUM(kolon): girilen kolondaki sayıların toplamını yollar



08221 Veri Tabanı II

T-SQL Fonksiyonlar

Scalar (tekil degerli) fonksiyonlar

Girilen degere gore bir deger ile calısır ve sonuc olarak bir deger yollar.



08221 Veri Tabanı II

T-SQL Fonksiyonlar

UCASE(c): bolgedeki karakterlerin hepsini buyuk yapar

LCASE(c): bolgedeki karakterlerin hepsini kucuk yapar

MID(c,start[,end]): yazı alanından karakterleri calıstırır

INSTR(c): yazı alanından karakterleri gosterir

LEFT(c,karakterNumarasi): text alanının girilen sayıya kadar olan kısmını yollar
(soldan sayar)

RIGHT(c,number_of_char): text alanının girilen sayıya kadar olan kısmını
yollar (sagdan sayar)



08221 Veri Tabanı II

T-SQL Fonksiyonlar

ROUND(c,hassasiyet): sayı alanını verilen hassasiyete gore yuvarlar

MOD(x,y): bolme işleminde kalanı gösterir (mod işlemi yapar)

NOW(): o anki sistem zamanını gösterir

FORMAT(c,format): alanın gosterim biçimini degiştirir

DATEDIFF(d,birinciTarih,ikinciTarih): tarih hesaplarını yapmak için kullanılır

08221 Veri Tabanı II

Örnek Tablolar ve Sorgular

ogrenci tablosundaki alan isimleri ve örnek veriler;

<u>ogr_no</u>	<u>adi</u>	<u>soyadi</u>	<u>cinsiyeti</u>
88050071	Muhittin	Altinkaya	E
89050020	Selda	Anar	K
91050099	Yahya	Demircan	E

08221 Veri Tabanı II

Notlar tablosundaki alan isimleri ve örnek veriler;

<u>ogr_no</u>	<u>kodu</u>	<u>donem</u>	<u>sube</u>	<u>vize</u>	<u>final</u>	<u>basari</u>
88050071	A308	2003-04/2	A	45	60	60
88050071	A401	2003-04/1	B	50	94	94
89050020	A308	2003-04/2	A	50	50	50
89050020	A307	2003-04/1	A	80	40	45
91050099	A308	2003-04/2	A	20	60	50
91050099	A308	2003-04/2	B	60	60	66
91050099	A401	2003-04/1	A	80	80	94
91050099	A445	2003-04/1	A	0	0	0
91050099	A445	2003-04/1	A	60	60	66



08221 Veri Tabanı II

Örneklere kullanılacak SQL'de kullanılan fonksiyonları;

SUM(DISTINCT alan_adi) Belirtilen alan için toplam fonksiyonu (sayısal alanlarda)

COUNT(*) Toplam kayıt sayısı

COUNT(DISTINCT alan_adi) Belirtilen alana göre farklı olan kayıtların sayısı

MAX([DISTINCT] alan_adi) Belirtilen alanda en büyük değer

MIN(DISTINCT alan_adi) Belirtilen alanda en küçük değer

AVG([DISTINCT] alan_adi) Belirtilen alanda ortalama değeri.

DISTINCT parametresi ile belirtilen alanda bulunan bilgilerden aynı olanlarından sadece birinin seçilmesini sağlar.



08221 Veri Tabanı II

Örnek 1:

```
SELECT COUNT(*) FROM notlar.
```

Sonucunda ekrana gelecek kayıt sayısı dur.



08221 Veri Tabanı II

Örnek 1:

```
SELECT COUNT(*) FROM notlar.
```

Sonucunda ekrana gelecek kayıt sayısı 9 dur.

Notlar tablosunda 9 adet kayıt vardır.



08221 Veri Tabanı II

Örnek 2:

```
SELECT COUNT(DISTINCT ogr_no) FROM notlar.
```

Sql ifadesinde ekrana gelecek kayıt sayısı 3 olacaktır.



08221 Veri Tabanı II

Örnek 3:

SELECT MAX(basari) FROM notlar.

En büyük basari notunu, ekrana 94 olarak görüntüler.

En büyük değer kaç tane olursa olsun bir tanesi görüntülenir.



08221 Veri Tabanı II

Örnek 4:

```
SELECT COUNT(*) FROM ogrenci
```

ogrenci tablosundaki kayıt sayısını 3 olarak verecektir.



08221 Veri Tabanı II

SORGULAMA

SELECT [ALL|DISTINCT] { * | alan_adi_listesi }

[INTO deęişken_listesi]

FROM tablo-ismi

[WHERE koşul]

[GROUP BY alan_adi1 [, alan_adi2] ...]

[HAVING search-condition]

[ORDER BY siralama_alanlari]

şeklinde genel bir ifadesini bir kez daha hatırlayalım.



08221 Veri Tabanı II

Herhangi bir şart kullanmadan kayıtları sorgulama:

Örnek 1:

SELECT adi, soyadi FROM ogrenciler

ogrenciler tablosundaki tüm öğrencilerin adını ve soyadını ekrana görüntüleyecektir.



08221 Veri Tabanı II

Örnek 2:

SELECT vize*10 FROM notlar.

Notlar tablosundaki tüm öğrencilerin vize notunu 10 ile çarpımını ekrana görüntüleyecektir.

Burada vizelerin 10 ile çarpımı sadece ekrana görüntülenecek tabloda ise değişmeyecektir.



08221 Veri Tabanı II

Örnek 3:

```
SELECT SUM(basari) FROM notlar.
```

Öğrencilerin başarı notlarının toplamını ekrana yazdıracaktır.



08221 Veri Tabanı II

Örnek 4:

SELECT adi + soyadi FORMAT "x(25)" FROM ogrenciler.

Öğrencilerin adı ve soyadını bitişik yazan SQL ifadesidir.

Bu durumda iki alanın birleşmesi sonucunda karakter sayısı artacağı için yeni bir format belirtmek gerekir.

Karakter alanlarda varsayılan görüntüleme 8 karakterdi, burada iki bilinin birleşmesi yeni bir bilgi alanı gibi davranacağından format belirtmek gereklidir.



08221 Veri Tabanı II

Örnek 5:

SELECT * FROM notlar.

Notlar tablosundaki bütün bilgi alanlarının içeriklerini ekrana görüntüleyecektir (* joker gibi kullanılmaktadır, "*" ile kullanımda select ile from arasına başka ifade yazılamaz).



08221 Veri Tabanı II

Örnek 6:

```
DEF VAR sayi AS INT
```

```
SELECT COUNT(*) INTO sayi FROM notlar.
```

Şeklinde bir kullanımla kayıt sayısı, “sayi” değişkeni içine aktarılmış olur, böylece elde edilen kayıt sayısı programın devamında amaca göre kullanılabilir.

Burada into parametresi kullanıldığı için sql sonucunda ekran hiçbir görüntü gelmeyecektir.



08221 Veri Tabanı II

Örnek 7:

```
DEF VAR ksayisi AS INT.
```

```
DEF VAR ntoplam AS INT.
```

```
SELECT COUNT(*),SUM(basari) INTO ksayisi,ntoplam
```

```
FROM notlar.
```

Kayıt sayısı “ksayisi” değişkenine, basari notlarının toplamı da “ntoplam” değişkeni içine aktarılmış olacaktır.



08221 Veri Tabanı II

Tek kayıt sorgulama:

Herhangi bir tabloya ait kayıtlardan birer tanesinin listesini almak için;

Örnek 1:

SELECT DISTINCT kodu FROM notlar.

Sorgu sonucu listelenecek kayıtlar;

Kodu A308, A401, A307, A445



08221 Veri Tabanı II

Tek kayıt sorgulama:

Herhangi bir tabloya ait kayıtlardan birer tanesinin listesini almak için;

Örnek 1:

SELECT DISTINCT kodu FROM notlar.

Sorgu sonucu listelenecek kayıtlar;

Kodu A308, A401, A307, A445



08221 Veri Tabanı II

Sorgu sonucu dönen kayıtları sıralama:

Herhangi bir tablodaki kayıtları herhangi bir alan adına göre azalan (artan) sırada sıralayabilmek için;

SELECT

FROM

ORDER BY [,alan_listesi] DESC | ASC.

ifadesi kullanılır (varsayılan olduğu için artan sırada olacaksa ASC kullanılmaz).



08221 Veri Tabanı II

Örnek 1:

```
SELECT ogr_no, adi, soyadi FROM ogrenciler  
  
ORDER BY soyadi DESC.
```

Öğrencilerin numara , ad ve soyadlarını , soyadı sıralı olarak Z'den A'ya sıralayarak ekrana görüntüler.



08221 Veri Tabanı II

Örnek 2:

```
SELECT ogr_no, basari FROM notlar ORDER BY basari DESC.
```

Öğrencinin numara ve başarı notlarını, başarı notuna göre azalan sırada ekran görüntüler.



08221 Veri Tabanı II

Örnek 3:

```
SELECT ogr_no, basari FROM notlar ORDER BY basari  
DESC,ogr_no DESC.
```

Öğrencinin numara ve başarı notlarını, başarı notuna göre azalan sırada listelerken notu aynı olanları da numarasına göre azalan sırada ekranda görüntüler.



08221 Veri Tabanı II

Şartlı kayıt sorgulama:

Bir tabloda istenilen şarta uygun kayıtların listesi için;

SELECT FROM WHERE

ifadesi kullanılır.

Örnek 1:

SELECT adi, soyadi, ogr_no FROM ogrenciler WHERE soyadı = " " .

Ogrenciler tablosundaki soyadı boş olan kayıtların ad,soyad ve numaralarını ekranda görüntüler.



08221 Veri Tabanı II

Örnek 2:

```
SELECT ogr_no, basari FROM notlar WHERE basari = 100.
```

Notlar tablosundaki başarı notu 100 olan öğrencilerin numaralarını ve başarı notlarını listeler.



08221 Veri Tabanı II

Boş ve dolu alanlara göre sorgu:

Boş alanları listelemek için;

SELECT FROM WHERE IS NULL.

Örnek 1:

SELECT ogr_no, adi, soyadi FROM ogrenciler

WHERE cinsiyeti IS NULL.

Cinsiyet alanı boş olan öğrencilerin numaralarını, ad ve soyadlarını listeler.



08221 Veri Tabanı II

Boş olmayan alanları listelemek için;

SELECT FROM

WHERE IS NOT NULL.

Örnek 2:

SELECT ogr_no, adi, soyadi FROM ogrenciler

WHERE cinsiyeti IS NOT NULL.

Cinsiyet alanı boş olmayan olan öğrencilerin numaralarını, ad ve soyadlarını listeler.



08221 Veri Tabanı II

Tam olmayan (eksik) bilgiye göre sorgu:

İstenilen değer ile başlayan kayıtları listelemek için;

SELECT FROM

WHERE LIKE “%”.

Örnek 1:

SELECT adi,soyadi FROM ogrenciler WHERE adi LIKE “A%”.

Adı “A”..... ile başlayan kayıtların adı ve soyadını listeler.



08221 Veri Tabanı II

İstenilen değer ile biten kayıtları listelemek için;

SELECT FROM

WHERE LIKE “%”.

Örnek 2:

SELECT adi,soyadi FROM ogrenciler WHERE adi LIKE “%A”.

Adı “A” ile biten kayıtların adı ve soyadını listeler.

08221 Veri Tabanı II

İçinde herhangi bir yerde istenilen değer geçen kayıtları listelemek için;

SELECT FROM

WHERE LIKE “%%”.

Örnek 3:

SELECT adi,soyadi FROM ogrenciler WHERE adi LIKE “%HY%”.

Adı içinde “HY” geçen kayıtların adı ve soyadını listeler.



08221 Veri Tabanı II

İki Tabloyu Beraber Sorgulama (Tabloya Alias [Geçici İsim] Verme):

SELECT FROM tablo1,tablo2 WHERE koşullar.

Birden fazla tablodan aynı anda bilgi getirilmesi gerektiğinde, ortak alanlar üzerinden birleştirme işlemleri yapılır. Birleştirme işlemi koşullar bölümünde yapılır, ortak olan alanlar eşleştirilir.



08221 Veri Tabanı II

İki Tabloyu Beraber Sorgulama (Tabloya Alias [Geçici İsim] Verme):

Örnek 1:

```
SELECT o.adi, o.soyadi, o.ogr_no, n.basari
```

```
FROM ogrenciler o,notlar n
```

```
WHERE o.ogr_no=n.ogr_no and n.kodu='A308'.
```

A308 dersinde final notu 100 olan öğrencilerin adlarını, soyadlarını, numaralarını ve başarı notlarını listeler. Burada “o” ogrenciler tablosuna, “n” notlar tablosuna verilmiş geçici isimlerdir. Bu geçici isimler sayesinde sql i daha anlaşılır ve kısa yazmak mümkündür.



08221 Veri Tabanı II

Örnek 2:

```
SELECT DISTINCT ogr_no, adi, soyadi
```

```
FROM ogrenciler o, notlar n
```

```
WHERE o.ogr_no = n.ogr_no AND n.kodu = "A206".
```

A206 dersini alan tüm öğrencilerin numaralarını, ad ve soyadlarını listeler.



08221 Veri Tabanı II

Kayıtları Grublama:

Kayıtları, belirli alanlara göre grubluyarak sorgu yapılmasını sağlar.

Örnek 1:

```
SELECT kodu,count(*) FROM notlar WHERE ogr_no = "91050099" GROUP BY  
ogr_no,kodu.
```

Bu örnekte 91050099 nolu öğrencinin, aldığı dersleri ve bu dersleri kaç kez aldığını listeler.



08221 Veri Tabanı II

Kayıtları Gruplama:

Kayıtları, belirli alanlara göre gruplayarak sorgu yapılmasını sağlar.

Örnek 1:

```
SELECT kodu,count(*) FROM notlar WHERE ogr_no = "91050099" GROUP BY  
ogr_no,kodu.
```

Bu örnekte 91050099 nolu öğrencinin, aldığı dersleri ve bu dersleri kaç kez aldığını listeler.



08221 Veri Tabanı II

UYGULAMA :

MSSQL kullanarak,

Örnekte olduğu gibi iki tablo hazırlayınız.

Yansılarda gösterildiği şekli ile verileri giriniz. Sorgulamaları yapınız.

08221 Veri Tabanı II



DEVAMI HAFTAYA