

HİTİT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

Veri Tabanı Yönetim
Sistemleri

İÇİNDEKİLER

1. Veri Tabanı Temel Kavramları <i>Doç. Dr. ABDULKADİR ÖZDEMİR</i>	4
2. Veri Tabanı ve Normalizasyon <i>Doç. Dr. ABDULKADİR ÖZDEMİR</i>	24
3. Veri Tabanı Araçlarının Kurulumunu Yapmak <i>Dr. Öğr. Üyesi SERDAR AYDIN</i>	43
4. Tabloları Oluşturmak ve Özelliklerini Belirlemek <i>Dr. Öğr. Üyesi SERDAR AYDIN</i>	67
5. TSQL ile Veri Tabanı ve Tabloları Oluşturmak, Özelliklerini Belirlemek <i>Dr. Öğr. Üyesi SERDAR AYDIN</i>	91
6. Sorgu Oluşturmak ve Çeşitlerini Kullanmak-1 <i>Dr. Öğr. Üyesi SNAN KUL</i>	112
7. Sorgu Oluşturmak ve Çeşitlerini Kullanmak-2 <i>Dr. Öğr. Üyesi SNAN KUL</i>	136
8. Sorgu Oluşturmak ve Çeşitlerini Kullanmak - 3 <i>Dr. Öğr. Üyesi SNAN KUL</i>	157
9. Sorgu Oluşturmak ve Çeşitlerini Kullanmak - 4 <i>Ar. Gör. YAKUP BAYOĞLU</i>	180
10. İlişkili Tablolar ile Sorgu Hazırlamak <i>Ar. Gör. YAKUP BAYOĞLU</i>	203
11. Görüntü (VIEW), Store Prosedür ve Fonksiyonlar <i>Ar. Gör. YAKUP BAYOĞLU</i>	223
12. Veri Tabanı Yönetimi Yapmak <i>Dr. Öğr. Üyesi AHMET KAMİL KABAKU</i>	248
13. Veri Tabanı Güvenliğini Sağlamak <i>Dr. Öğr. Üyesi AHMET KAMİL KABAKU</i>	271
14. Veri Tabanında Yedekleme ve Geri Yükleme <i>Dr. Öğr. Üyesi AHMET KAMİL KABAKU</i>	291

SORGU OLUŐTURMAK VE ÇEŐİTLERİNİ KULLANMAK-2



- Verileri gruplayarak analiz etme
- Grup fonksiyonları (max, min, count, avg, sum)
- Birden fazla sütuna göre gruplama
- Grup koşullarının kullanımı

İÇİNDEKİLER



Atatürk Üniversitesi
Açıköğretim Fakültesi

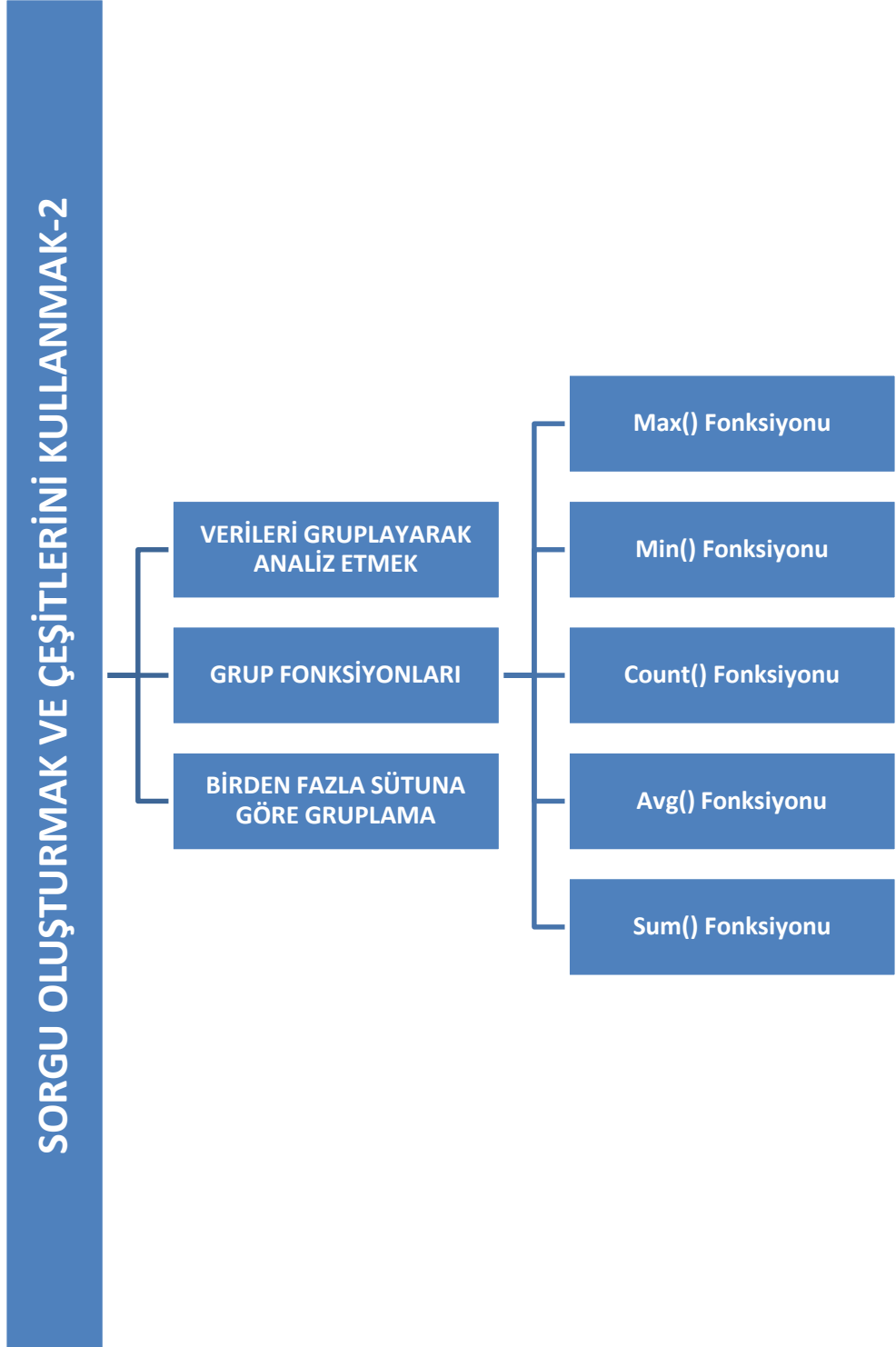
**VERİ TABANI
YÖNETİM SİSTEMLERİ**
Dr. Öğr. Üyesi Sinan KUL



- Bu üniteyi çalıştıktan sonra;
 - Grublamanın ne olduğunu ve grublamanın nasıl yapıldığını kavrayabilecek,
 - Grublama fonksiyonlarının işlevlerini ve fonksiyonların kullanımlarını öğrenebilecek,
 - Birden fazla sütuna göre gruplama ve gruplanmış veriler ve grup fonksiyonları üzerinde koşul tanımlama izlemlerini gerçekleştirebilecek,
 - Null değerlerinin gruplamaya etkisi hakkında bilgi edinebileceksiniz.

HEDEFLER

ÜNİTE
7



GİRİŞ

Önceki üniteye tabloların sütun eklenmesi ve var olan sütunlarının özelliklerinin güncellenmesi konularına değinildikten sonra SQL dilinde en çok kullanılan “Select” deyiminin kullanımına giriş yapılmıştı. “Select” deyiminin from deyimi ile kullanılarak veri tabanı tabloları veya görünümülerinden kayıtların nasıl çekildiği gösterilmişti. Yıldız operatörü ile kullanılarak ilgili tablodaki tüm sütunlar görüntülenebilirken sadece görüntülenmek istenen sütun verilerinin listelenebilmesi için de “Select” deyiminden sonra sadece ilgili sütunların yazılması gerektiği bilgisi verilmişti.

“Select” deyimini kullanarak veri tabanı tablolarındaki tüm kayıtların veya özel şartlara uyan kayıtların ilgili sütunlarının nasıl listelenebildiği gösterilmişti. Kayıtların filtrelenmesinde kullanılan karşılaştırma operatörleri tanıtarak her biri için açıklayıcı örnekler verilmişti. Önceki üniteye son olarak verilerin tablolardan sıralanarak listelenebilmesi için “order by” operatörünün nasıl kullanılabileceği ve sorgu sonucu dönen ilk N adet kaydın nasıl listelenebileceği gösterilmişti.

Bu üniteye ise verilerin istatistiksel analiz için nasıl gruplanacağı ve gruplanmış veriler üzerinde bazı hesaplama (gruplama) fonksiyonlarının nasıl işleteceği konuları işlenecektir. Gruplama fonksiyonlarının kullanım kuralları işlenirken her bir fonksiyonun parametre olarak alabileceği veri türüne değinilecektir. Bazı fonksiyonlar (Sum, Avg) sadece sayısal veriler için çalışabilirken diğerleri (Max, Min) metinsel ve tarihsel veriler için çalışabilmektedir. Count() fonksiyonu ise tüm veri türleriyle çalışabilmektedir. Count() fonksiyonunun diğer gruplama fonksiyonlarından farklı olduğu diğer bir konu ise parametre olarak “*” (yıldız) karakterini de alabilmesidir.

Gruplama fonksiyonlarının sütun adını parametre olarak kullandığı durumda “Distinct” ve “All” deyimlerinden biri sütun adından önce tercihen kullanılabilir. Varsayılan olarak “All” deyimini kabul edilmektedir. Yani herhangi bir deyim kullanılmadığında tüm değerler hesaplama katılmaktadır. Distinct kullanıldığında ise aynı olan değerler sadece bir kere hesaplama katılmaktadır.

“Select” cümlesi içinde koşul belirtilirken, yani satırlar filtrelenirken, where operatöründen faydalandığı daha önceki üniteye öğretilmişti. Ancak koşul ifadesi içinde gruplama fonksiyonundan dönen değer kullanılacaksa, bu koşul ifadesinin having anahtar sözcüğünden sonra kullanılması gerekmektedir. Üniteye ayrıca, “select” cümlesi kurulurken yazım kurallarından bahsedilecek ve anahtar deyimlerin hangi sırayla cümle içinde yer alması gerektiği kurallarına değinilecektir.

Veriler gruplanmadan da gruplama fonksiyonlarının kullanılabilirliği ki bu durum için üniteye çeşitli örnekler verilecektir. Üniteye, verilerin birden fazla alana göre nasıl gruplanacağı ve gruplu verilerin nasıl görüntülenebileceğine de değinildikten sonra son olarak gruplama fonksiyonlarından dönen değerler üzerinden nasıl koşul tanımlanacağı detaylı olarak incelenecektir.

VERİLERİ GRUPLAYARAK ANALİZ ETMEK

Veriler belirli özelliklerine göre istatistiksel olarak analiz edilebilmesi için kendi aralarında gruplanarak listelenebilmektedir. Gruplama ile genellikle grup toplamı, grup ortalaması, grup maksimumu, grup minimumu ve grup sayısı gibi değerlere ulaşılmak istenmektedir.



“Group By” deyimini “Where” operatöründen sonra ve “Order By” operatöründen önce gelmektedir.

Verilerin gruplanmasında yani verilerin bir veya birden fazla tablodan gruplanarak sorgulanması için yazılan “Select” cümlesi içinde “Group By” deyimini kullanılmaktadır. “Group By” deyiminin “Select” yapısı içindeki tam konumu “Where” operatöründen sonra ve “Order By” operatöründen öncedir (Gözüdeli, 2010). Yani veriler gruplanmadan önce filtrelenecekse “Where” deyimini “Group by” deyiminden önce kullanılmalıdır. Aynı şekilde gruplanan veriler sıralanarak listelenmek istenirse de sıralama ifadesi olan “Order By”, “Select” cümlesinin sonunda gelmelidir.

İçinde gruplamanın bulunduğu sorgu yapılarında sıklıkla yapılan hatalardan biri gruplama işlemine tabi tutulmayan alanların da görüntülenmek istenmesidir. Yani eğer gruplama operatörü kullanılmışsa sadece gruplanan alanlar ve gruplama fonksiyonlarından (Gruplama fonksiyonları bir sonraki konu başlığında işlenecektir.) dönen değerler görüntülenebilmektedir. Gruplanan alanlardaki veriler ise ilgili tablolardan benzersiz (eşsiz) olarak çekilmektedir.

Diğer bir dikkat edilecek konu, “Group By” ve “Order By” ifadelerinin birlikte kullanımları durumunda “Order By” ile birlikte kullanılan bütün sütun isimlerinin “Group By” ile gruplandırılmasının zorunluluğudur. Yani gruplama işlemi varsa görüntülenen ve sıralamaya sokulan bütün alanların gruplamaya alınması gerekmektedir.

Örnek olarak “Birey” tablosunda girilmiş olan “DogumYeri” verilerini benzersiz olarak listeleyelim (Şekil 7.1). “Group By” deyiminden sonra “DogumYeri” ifadesini yazarak sorgu sonucu dönen kayıtlar, “DogumYeri” alanına göre gruplanmaktadır.



Örnek

```
•SELECT DogumYeri FROM Birey GROUP BY DogumYeri
```



Gruplanan veriler benzersiz olarak listelenmektedir.

```

SELECT
  DogumYeri
FROM
  Birey
GROUP BY
  DogumYeri

```

	DogumYeri
1	Bayburt
2	Bitlis
3	Edime
4	Erzurum
5	Gümüşhane
6	Hatay
7	İstanbul
8	Karabük
9	Kars

Şekil 7.1. Birey Tablosunun “Dogumyeri” Alanına Göre Gruplanarak Listelenmesi (21 Kayıt)

Gruplanan veriler ayrıca, sıralanmak istenirse sorgunun sonunda “Order By” deyimini yazılmalıdır. Şekil 7.2.deki örnek, sorgu sonucu dönen kayıtların “DogumYeri” alanına göre nasıl sıralanacağını göstermektedir.

Örnek

•SELECT DogumYeri FROM Birey GROUP BY DogumYeri ORDER BY DogumYeri



“Order By” deyimini sorgunun en altında yer almaktadır.

```

SELECT
  DogumYeri
FROM
  Birey
GROUP BY
  DogumYeri
ORDER BY
  DogumYeri

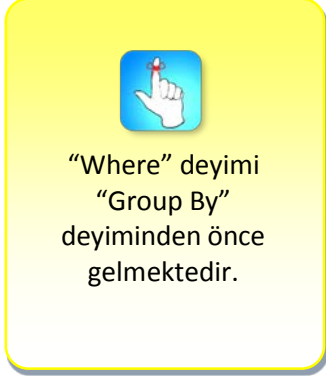
```

	DogumYeri
1	Bayburt
2	Bitlis
3	Edime
4	Erzurum
5	Gümüşhane
6	Hatay
7	İstanbul

Şekil 7.2. “Dogumyeri” Alanına Göre Gruplandıktan Sonra Sıralanarak Listelenmesi (21 Kayıt)

Şimdi sorgumuzu biraz daha karmaşıktırarak kayıtları gruplamadan önce bir koşul ekleyelim ve filtrelenen veriler üzerinde gruplama yapalım. Sonrasında ise gruplu verileri sıralayarak listeleyelim.

Where” deyimini ile tanımlayacağımız koşulda “cinsiyeti” bilgisi “Erkek” olan bireyler çekildikten sonra verileri, “DogumYeri” alanına göre gruplayarak “DogumYeri” alanına göre alfabetik olarak (A’dan Z’ye) sıralayalım (Şekil 7.3).



```

SQLQuery1.sql - Sin...(SINAN\sinan (54))* x
Select
  DogumYeri
from
  Birey
where
  Cinsiyeti = 'Erkek'
Group By
  DogumYeri
Order By
  DogumYeri
  
```

DogumYeri
1
2
3
4
5
6

N:\sinan (54) | OgrrenciBilgi | 00:00:00 | 18 rows

Şekil 7.3. Erkek Bireylerin Doğum Yerlerine Göre Gruplanması Ve Sıralanması (18 Kayıt)

Bu örnek ile sorguya tanımlanan koşulun (“Where”), grupta ifadesinden (“Group BY”) önce geldiğini ve sıralama ifadesinin (“Order By”) SQL sorgu cümlesinin sonunda geldiğini görmüş olduk.

GRUP FONKSİYONLARI

Gruplama fonksiyonları bir sütundaki verileri işleme tabi tutarak tek bir değer döndürür. Gruplama fonksiyonlarıyla mesela grup üyelerinin sayısı alınabileceği gibi bir birimde çalışan bireylerin maaş ortalamasını almak için de gruplama fonksiyonları kullanılabilir. Bu bölüm altında detaylı olarak işlenecek olan gruplama fonksiyonlarından her bir grup için tek bir sonuç dönmektedir. Bu fonksiyonlar, “Max”, “Min”, “Avg”, “Sum” ve “Count” olmak üzere beş tanedir.

Gruplama fonksiyonları, parametre olarak aldıkları sütundaki “null” olmayan değerler üzerinden işlem yapmaktadır. Diğer bir deyişle “null” değerleri görmezden gelmektedir (Adar, 2012). Örneğin “personel” tablosunda “maaş” alanının toplam bilgisi görüntülenmek istendiğinde, ilgili alanda “null” olmayan kayıtlar için işlem yapılmaktadır.

Veriler gruplanmadığında yani “Group By” ifadesi kullanılmadığında tüm veriler tek bir grup içinde kabul edilir. Dolayısıyla gruplanmayan veriler üzerinde de grup fonksiyonları kullanılabilir. Ancak gruplama fonksiyonları kullanılırken fonksiyon harici bir sütun değeri görüntülenmek istenirse bu sütuna göre verilerin gruplanması gerekmektedir. Aksi takdirde hata alınacaktır.

Grup fonksiyonlarının kullanım şekli aşağıdaki gibidir:

Söz Dizimi:

Fonksiyon([ALL | DISTINCT] ifade)

Tekrarlı kayıtları elemek için “DISTINCT” sözcüğünden faydalanılmaktadır. Böylece aynı veri fonksiyonda bir kere dikkate alınmaktadır.

“ALL” sözcüğü ise fonksiyonun bütün değerlere uygulanacağını söyler. “DISTINCT” ve “ALL” deyimleri kullanılmadığında SQL Server varsayılan olarak “ALL” seçeneğini kabul eder ve sorgudan dönen tüm değerler üzerinde işlem yapar. “DISTINCT” ve “ALL” deyimlerinin kullanımına ilişkin örnekler daha sonra verilecek, şimdilik burada keserek grup fonksiyonlarını tek tek açıklamaya geçelim.

Tablo 7.1. Grup Fonksiyonları

Fonksiyon	Desteklediği Veri Tipi	Açıklama
MAX	Sayısal, Metin, Tarih	Uygulandığı kolonun en büyük değerini verir.
MIN	Sayısal, Metin, Tarih	Uygulandığı kolonun en küçük değerini verir.
COUNT	Bütün veri tipleri	Koşula uyan kayıtların sayısını
AVG	Sayısal	Kolonun değerlerinin ortalamasını verir.
SUM	Sayısal	Kolonun değerlerinin toplamını verir.



“Sum” ve “Avg” fonksiyonları yalnızca sayısal değerler içeren sütunlar için kullanılır.

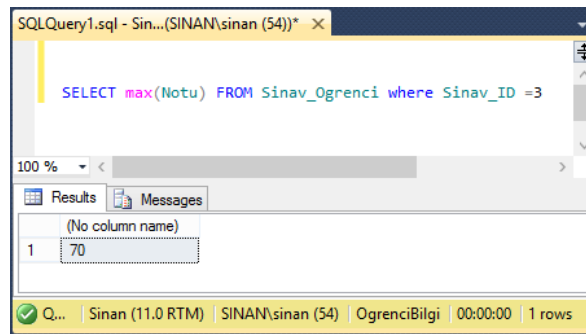
Max() Fonksiyonu

Sorgu sonucu dönen kayıtlar içinde, “Max” fonksiyonu, parametre olarak aldığı sütun içindeki en büyük değeri bulur. Max fonksiyonu parametre olarak sayısal, metinsel ve tarihsel veri türlerindeki sütun adlarını veya değerleri almaktadır (Elbahadır, 2012).

Fonksiyonun daha iyi anlaşılabilmesi için bir örnek verelim. Örneğimizde (Şekil 7.4) öğrencilerin bir sınavdan (3 nolu sınav) aldıkları en yüksek notu görüntüleyelim. Hangi sınava ait kayıtlarla ilgilendiğimizi “Where” ifadesi ile belirtelim (Where Sınav_Id=3) . En yüksek sınav notu bilgisine erişmek ve görüntülemek içinse “MAX(Notu)” ifadesini “Select” deyiminden sonra yazalım.

Örnek

```
•SELECT MAX(Notu) FROM Sınav_Ogrenci WHERE Sınav_ID =3
```



Şekil 7.4. 3 Nolu Sınavda Alınan En Yüksek Notun Görüntülenmesi

Sorgumuzda “Group By” deyimine yer vermedik, grupta yapmadan da

gruplama fonksiyonlarının kullanılabilceğini göstermiş olalım istedik. “Group By” deyimi kullanılmadığı için veriler tek grup olarak kabul edilmektedir ve sorgu sonucu tek kayıt dönmektedir.

Şimdi de notları sınavlara göre gruplandırarak her bir sınavdan alınan en yüksek notları görüntüleyelim (Şekil 7.5). Gruplama fonksiyonu içinde geçmeyen her bir sütun adının “Group By” deyimi ile kullanılması gerektiğini daha önce söylemiştik. Örneğimizde “Group By Sınav_ID” ifadesi ile sorgu sonucu dönen kayıtların “Sınav_ID” alanına göre gruplanacağını belirtmektedir. “Sınav_ID” verisinin görüntülenmesi veya görüntülenme sırası tercihe bağlıdır. Yani “Select” deyiminden sonra hiç kullanılmayabilir veya “MAX(Notu)” ifadesinden önce veya sonra kullanılabilir.

The screenshot shows a SQL query window with the following query: `Select Sınav_ID, EnYüksekNot=max(Notu) from Sınav_Ogrenci group by Sınav_ID`. The results are displayed in a table with 8 rows.

Sınav_ID	EnYüksekNot
1	3
2	4
3	5
4	6
5	8
6	9
7	10
8	45

Şekil 7.5. Sınavlarda Alınan En Yüksek Notların Görüntülenmesi (8 Kayıt)

Örnekte, “Max” fonksiyonuna “EnYüksekNot” takma adı verilmiş.

Örnek

- SELECT Sınav_ID, EnYüksekNot=MAX(Notu) FROM Sınav_Ogrenci GROUP BY Sınav_ID

Min() Fonksiyonu

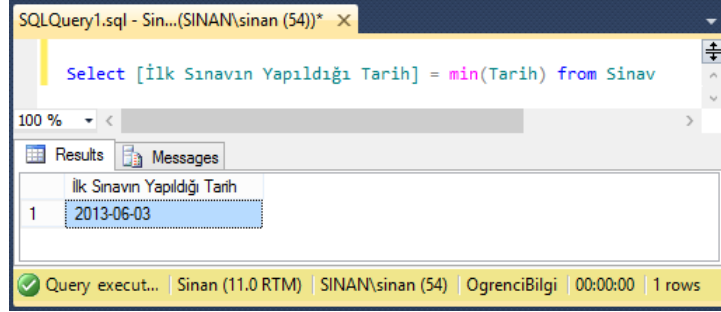
“Min” fonksiyonu, “Max” fonksiyonuyla aynı kullanım özellikleri göstermektedir. Farkı, parametre olarak aldığı sütun içindeki en küçük değeri bulmasıdır. “Max” fonksiyonu gibi “Min” fonksiyonu da sayısal, metinsel ve tarihsel veri türlerindeki sütun adlarını veya değerleri parametre olarak almaktadır. Yani sütundaki en küçük tarihi (en eski tarihi), sayıyı veya metni (alfabetik olarak sıralandığında en üstte kalan değeri) döndürmektedir.

Örnek olarak ilk yapılan sınavın tarihini bulalım. Bunun için “Sınav” tablosundaki “Tarih” alanından faydalanacağız. Fonksiyon sonucu dönen değere, “İlk Sınavın Yapıldığı Tarih” takma adını verelim. Takma ad içinde boşluk karakterleri bulunduğu için köşeli parantezler içinde kullanıldığını dikkat ediniz (Şekil 7.6).



Örnek

•SELECT [İlk Sınavın Yapıldığı Tarih] = MIN(Tarih) FROM Sınav



Şekil 7.6. En Önce Yapılan Sınavın Tarihinin Görüntülenmesi

“Min” fonksiyonunun, tarih veri türünde bir alan ile kullanımında en eski veriye ulaşıldığını görmüş olduk. Tarih veri türündeki alan, “Max” fonksiyonu ile kullanıldığında ise en güncel verinin yani en büyük tarih verisinin görüntüleneceğini söyleyebiliriz.

Count() Fonksiyonu

“Count” fonksiyonu en genel anlamıyla kayıt sayısını (tam sayı türünde) döndürmektedir. Parametre olarak “*” (yıldız) simgesi kullanıldığında sorgudan dönen kayıt sayısını verirken parametre olarak sütun adı kullanıldığında ise ilgili sütunda “null” olmayan kayıt sayısını döndürür. Herhangi bir sütunda farklı olan değer sayısını görüntülemek için de ilgili sütun adı “Distinct” deymi ile birlikte kullanılabilir.

“Count” fonksiyonu da diğer gruplama fonksiyonları gibi grupsuz veriler için yani group by deymi olmayan bir select cümlesi ile listelenen kayıtlar için kullanılabilir. “Count” fonksiyonunu diğer gruplama fonksiyonlarından ayıran özellik ise bu fonksiyonun, “*” parametresiyle çalışabilen tek gruplama fonksiyonu olmasıdır.

“Count” fonksiyonunun kullanımına örnek olması için “Birey” tablosundaki kayıt sayısını getiren sorguyu yazalım. Bu sorguda herhangi bir şart belirtmeyerek tablodaki kayıt sayısının tamamını alalım. Fonksiyon sonucu dönen değeri, “sayı” takma adı ile görüntüleyelim.

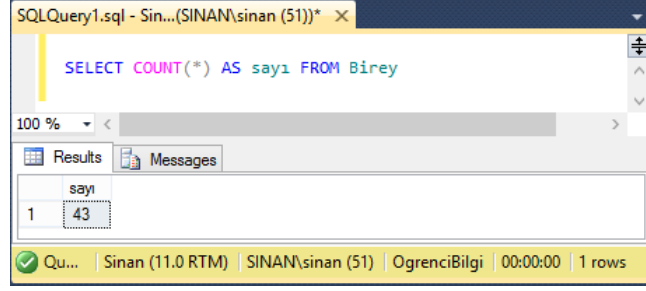


Örnek

•SELECT COUNT(*) AS sayı FROM Birey



“Count” fonksiyonu “*” parametresiyle çalışabilen tek gruplama fonksiyonudur.



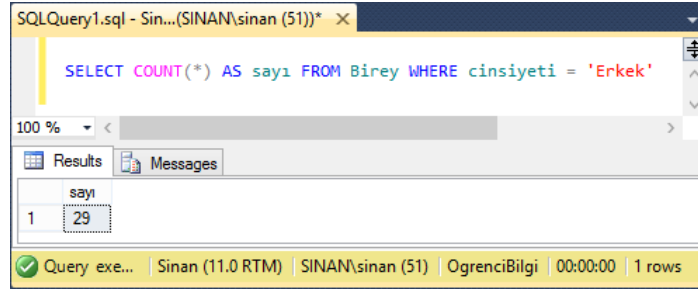
Şekil 7.7. Birey Tablosundaki Kayıt Sayısı

Şimdi de filtrelenmiş veriler üzerinde grup fonksiyonunu çağırılm ve Birey tablosunda “Cinsiyeti” bilgisi “Erkek” olan kayıtların sayısını veren sorguyu yazalım (Şekil 7.7). Bir önceki sorgudan (Şekil 7.6) tek farkı sorgu sonunda yer alan ve “Where” deyimi ile bir koşul tanımlanmasıdır.



Örnek

- SELECT COUNT(*) AS sayı FROM Birey WHERE cinsiyeti = 'Erkek'



Şekil 7.8. Birey Tablosundaki “Cinsiyeti” “Erkek” Olan Kayıt Sayısı



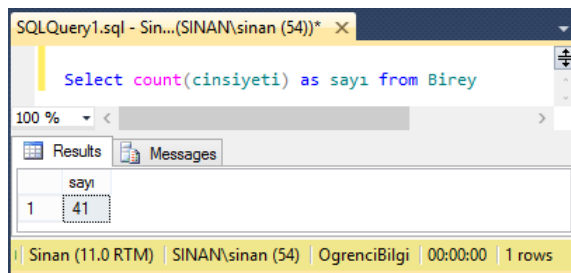
“Group By” deyimi olmadan gruptlama fonksiyonları kullanılırsa tek kayıt dönecektir.

Aşağıdaki SQL sorgusunda (Şekil 7.9) ise sütun isminin parametre olarak kullanılmasına örnek verilmiştir. Tanımlanan sorgu ile “Cinsiyeti” alanına değer girilmiş olan (“Null” olmayan) kayıt sayısı getirilmektedir.



Örnek

- SELECT COUNT(cinsiyeti) AS sayı FROM Birey

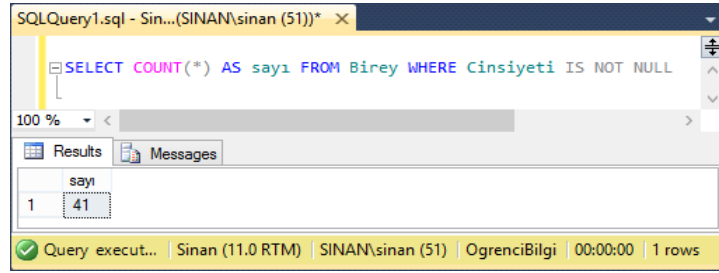


Şekil 7.9. Birey Tablosundaki “Cinsiyeti” Alanı Dolu Olan Kayıt Sayısı

“Cinsiyeti” bilgisi “Null” olmayan kayıt sayısına aşağıdaki sorgu (Şekil 7.10) ile de ulaşılabilir. “Where” deyiminden sonra “Cinsiyeti” alanının null olmaması koşulu tanımlanır ve sorgu sonucu dönen kayıt sayısı “Count” fonksiyonu ile alınır. “Count” fonksiyonunun * (yıldız) ile kullanıldığına dikkat ediniz.

Örnek

- SELECT COUNT(*) AS sayı FROM Birey WHERE Cinsiyeti IS NOT NULL

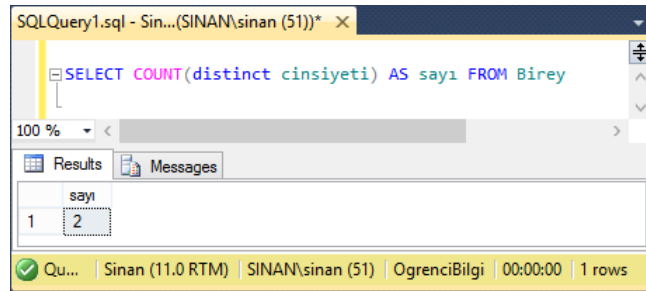


Şekil 7.10. Birey Tablosundaki “Cinsiyeti” Alanı Dolu Olan Kayıt Sayısı

Şimdi de “Distinct” deyiminin kullanımına da bir örnek olması için “Cinsiyeti” bilgisi aynı olan kayıtların bir kez sayılmasını sağlayan SQL kodunu yazalım (Şekil 7.11).

Örnek

- SELECT COUNT(DISTINCT cinsiyeti) AS sayı FROM Birey



Şekil 7.11. Birey Tablosundaki “Cinsiyeti” Alanı Dolu Olan Kayıt Sayısı

“Count” fonksiyonunu bu kez de gruplanan veriler ile kullanalım. Yani “Birey” tablosunda bulunan kayıtlardan “Cinsiyeti” alanı “Erkek” olan “Kız” olan ve Cinsiyeti alanına değer girilmeyen (null olan) kayıt sayısını bulan sorguyu yazalım (Şekil 7.12). Bunun için gruptandırma deyiminden (“Group By”) faydalanacağız. Bireyleri “Cinsiyeti” alanlarına göre gruptandıktan sonra kayıt sayısını görüntüleyelim.



Gruplama fonksiyonları yinelenen değerleri bir kere kullansın istenirse “Distinct” operatöründen faydalanılabilir.

Örnek

- SELECT Cinsiyeti, COUNT(*) AS sayı FROM Birey GROUP BY Cinsiyeti



Gruplamada
“Null” değerler
de
getirilmiştir.

Cinsiyeti	sayı	
1	NULL	2
2	Erkek	29
3	Kız	12

Şekil 7.12. Cinsiyet Bilgilerinin Kaç Defa Görüntülediğinin Listelenmesi. (3 Kayıt)

Şimdi de yukarıdaki örnekte geçen “Count(*)” ifadesini “Count(Cinsiyeti)” olarak değiştirelim ve sorgu sonucundaki farklılığı görelim (Şekil 7.13).

Cinsiyeti	sayı	
1	NULL	0
2	Erkek	29
3	Kız	12

Şekil 7.13. Cinsiyet Bilgilerinin Kaç Defa Geçtiğinin Listelenmesi.(Null Değerler Hariç)

Cinsiyeti bilgisi girilmemiş 2 adet birey kaydı bulunduğu hâlde örnekte de görüldüğü gibi “Count” fonksiyonu, parametre olarak sütun adı aldığı için Cinsiyeti null olanlar için kayıt sayısını 0 olarak görüntülemektedir.

Avg() Fonksiyonu

“Avg” fonksiyonu, parametre olarak aldığı değerlerin aritmetiksel ortalamasını almak için kullanılır. Bu fonksiyon sadece sayısal türdeki veriler ile çalışmaktadır. Diğer bir deyişle metinsel ve tarihsel veri türleri ile kullanıldığında hataya sebep olmaktadır.

Parametre olarak tam sayı veriler alırsa sonuç da tam sayı olur, parametre olarak noktalı sayı (float) veriler alırsa da sonuç noktalı sayı türünde bir sayı olacaktır. Metinsel ve tarihsel veri türündeki kolon adlarını ise parametre olarak almamaktadır.

“Avg” fonksiyonunun kullanımını anlamak için basit bir örnek verelim ve öğrencilerin belirli bir sınavdan (5 nolu sınavdan) aldıkları notların ortalamasını görüntüleyelim (Şekil 7.14).

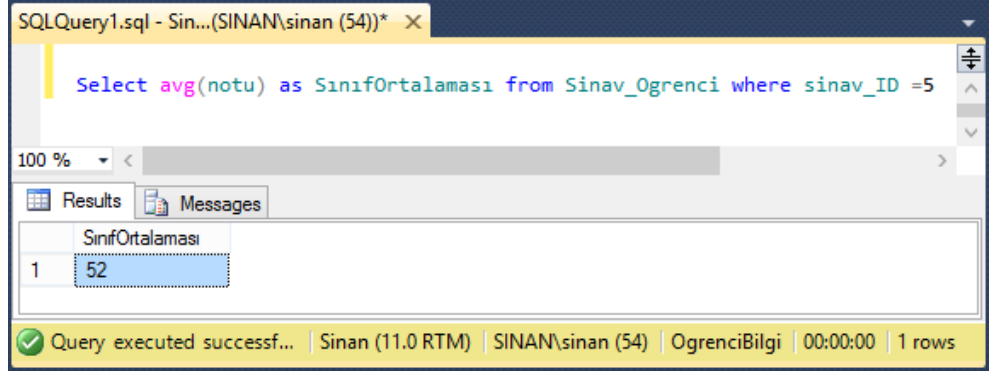


“Count” fonksiyonu
“null” olan sütun
değerleri için 0
değerini
döndürmektedir.



Örnek

•SELECT AVG(notu) AS SınıfOrtalaması FROM
Sınav_Ogrenci WHERE sınav_ID =5



Şekil 7.14. 5 Nolu Sınavdan Alınan Notların Ortalaması

Sum() Fonksiyonu

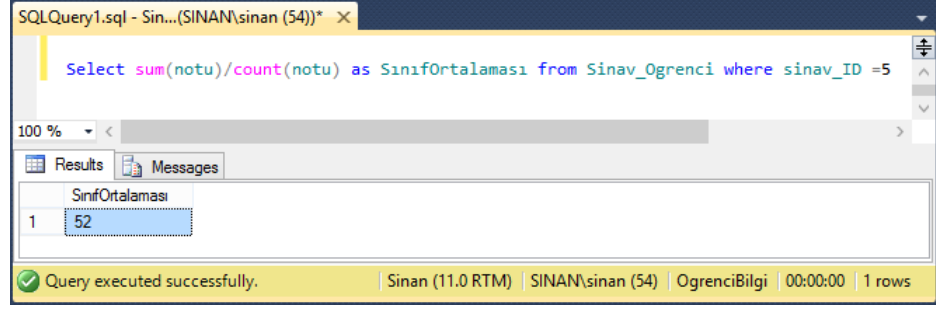
“Sum” fonksiyonu, parametre olarak aldığı sütundaki değerleri toplayarak sonucu geri döner. Kullanımı “Avg” fonksiyonunun kullanımına benzerdir. Bu fonksiyon da sayısal türde veriler için geçerlidir. Yani parametre olarak aldığı sütunda tutulan verilerin sayısal türde olması gerekmektedir. Parametre olarak tam sayı veriler alırsa sonuç da tam sayı olur; parametre olarak noktalı sayı (float) veriler alırsa da sonuç da noktalı sayı türünde bir sayı olacaktır. Metinsel ve tarihsel veri türündeki kolon adlarını ise parametre olarak almamaktadır.

“Sum” fonksiyonunun kullanımına örnek olması için, öğrencilerin bir sınavdan aldıkları notların ortalamasını, “Sum” ve “Count” fonksiyonlarını kullanarak hesaplayalım (Şekil 7.15). Ortalamayı bulurken ilgili sınavdan alınan puanların toplamı, sınava giren öğrenci sayısına bölünmektedir. Sınav notu girilen öğrenci sayısını, “Count” fonksiyonu ile bulabileceğimizi daha önce görmüştük. “Sum” fonksiyonu ile hesaplanan toplam değeri, “Count” fonksiyonu ile hesaplanan kayıt sayısı değerine bölüldüğünde ilgili sınavdan alınan notların ortalama değerine ulaşılmış olmaktadır.



Örnek

•SELECT SUM(notu)/COUNT(notu) AS SınıfOrtalaması
FROM Sınav_Ogrenci WHERE sınav_ID =5



Şekil 7.15. 5 Nolu Sınavdan Alınan Not Ortalaması

“Count” fonksiyonu tam sayı (“int”) türünde değer döndürdüğünü hatırlayalım. “Notu” alanı tam sayı (“int”) türünde veri içerdiği için de “Sum” ve “Avg” fonksiyonları da sonuç olarak tam sayı türünde veri döndürmektedir. Dolayısıyla SQL’de iki tam sayının birbirine bölünmesinin de yine bir tam sayı olarak hesaplanacağı bilgisi akıldan çıkarılmamalıdır.

BİRDEN FAZLA SÜTUNA GÖRE GRUPLAMA

Birden fazla sütuna göre gruplarken “Group By” operatöründen sonra ilgili sütun adları yazılmaktadır.

Yeri gelmişken daha önce öğrendiğimiz bir kuralı hatırlatarak bir örnek verelim. Gruplama yapılırken “Select” ifadesi ile görüntülenen bütün sütun adları, “Group By” deyimi ile birlikte yazılmalıdır demiştik. Yani görüntülenen ve/veya sıralamada kullanılan sütun adları gruplamada kullanılmış olmalıdır. Aksi takdirde hata mesajı alınacaktır. Örnekte (Şekil 7.16) “Birey” tablosundaki kayıtlar “DogumYeri” ve “Cinsiyeti” alanlarına göre gruplandıktan sonra her bir doğum yerine göre aynı cinsiyette olan birey sayıları görüntülenmektedir. Cinsiyet bilgisi bulunmayanlar ise elenmektedir. Ayrıca gruplanan veriler “DogumYeri” ve “Cinsiyeti” alanlarına göre sıralanmaktadır.



“Group By” deyimi ile gruplandırmaya katılmayan sütunlar “Select” deyimi ile kullanılamaz.



Örnek

- SELECT DogumYeri, Cinsiyeti, COUNT(*) AS Sayı FROM Birey WHERE Cinsiyeti is not null
- GROUP BY DogumYeri, Cinsiyeti
- ORDER BY DogumYeri, Cinsiyeti


```

Select
  DogumYeri, Cinsiyeti, Count(*) as Sayı
from
  Birey
Where
  Cinsiyeti is not null
Group By
  DogumYeri, Cinsiyeti
Order By
  DogumYeri, Cinsiyeti

```

	DogumYeri	Cinsiyeti	Sayı
1	Bayburt	Erkek	1
2	Bitlis	Erkek	2
3	Edirne	Erkek	1
4	Erzurum	Erkek	5
5	Erzurum	Kız	2
6	Gümüşhane	Erkek	2
7	Hatay	Erkek	1
8	Hatay	Kız	1

Şekil 7.16. Cinsiyet Bilgileri Girilmiş Olan Bireylerin Doğum Yerlerine Ve Cinsiyetlerine Göre Gruplanması Ve Sıralanarak Listelenmesi (27 Kayıt)

GRUP KOŞULLARININ KULLANIMI

Gruplama fonksiyonlarından dönen değerler üzerinden bir filtre tanımlanmak istendiğinde “Having” operatöründen faydalanılmaktadır. “Having” operatörünün “Select” cümlesindeki yeri, “Group By” operatöründen sonra ve “Order By” operatöründen ise öncedir. “Group By” operatörünün en genel kullanım şekli aşağıdaki gibidir:

Söz Dizimi:

```

SELECT Sütun_Adi1, Sütun_Adi2, ... , GruplamaFonksiyonu(Sütun_Adi)
FROM Tablo_Adi
WHERE Şartlar
GROUP BY Sütun_Adi1, Sütun_Adi2, ... HAVING
Gruplama_Fonksiyonu_Şartları ORDER BY Sıralama_İfadesi

```

Operatörlerin sorguda yazılış sırasının değiştirilmesi sorgunun çalışmamasına sebep olacaktır. “Select ... From ...” ifadesi ile getirilen kayıtlar için,

“Where ...” ifadesi ile şart tanımlanabildiğini görmüştük. “Group By ...” ifadesi ile gruplamaya tabi tutulan kayıtlar için gruplama fonksiyonları ile ilgili koşullar da “Having ...” ifadesi ile tanımlanmaktadır. Listelenecek kayıtlar ve kolonlar belirlendikten sonra istenirse kayıtlar üzerinde sıralama yapılabilmektedir. Dolayısıyla “Order By ...” ifadesi sorgu yazımında en son gelmektedir.



Gruplama fonksiyonlarından dönen değerler için bir koşul tanımlamak gerektiğinde “Having” operatörü kullanılmalıdır.



Örnek

- SELECT DogumYeri, Cinsiyeti, COUNT(*) AS Sayı FROM Birey
- GROUP BY DogumYeri, Cinsiyeti
- HAVING COUNT(*)>1
- ORDER BY DogumYeri, Cinsiyeti

```

Select
    DogumYeri, Cinsiyeti, Count(*) as Sayı
from
    Birey
Group By
    DogumYeri, Cinsiyeti
Having
    count(*)>1
Order By
    DogumYeri, Cinsiyeti

```

	DogumYeri	Cinsiyeti	Sayı
1	Bitlis	Erkek	2
2	Erzurum	Erkek	5
3	Erzurum	Kız	2
4	Gümüşhane	Erkek	2
5	İstanbul	Erkek	4
6	İstanbul	Kız	2
7	Muğla	Erkek	2
8	Sivas	Erkek	2

Şekil 7.17. Bireyler, Doğum Yerlerine Ve Cinsiyetlerine Göre Gruplandığında Gruptaki Kayıt Sayısı 1’den Büyük Olanların Sıralı Listelenmesi (9 Kayıt)

Şekil 7.17.deki örnekte bireyler, doğum yerlerine ve cinsiyetlerine göre gruplandıktan sonra gruptaki kayıt sayısı 1’den büyük olanlar listelenirken doğum yerlerine ve cinsiyetlerine göre artan şekilde sıralanmaktadır.

“Having” operatörünün kullanımıyla ilgili dikkat edilecek hususlardan biri de “Where” operatörüyle tanımlanan kısıtlar “Having” operatörü içinde de yer alabilirken “Having” operatörü içinde tanımlanması gereken kısıtların “Where” operatörü ile kullanılmasının hataya sebep olmasıdır. Gruplama fonksiyonları sonucu dönen değerler ile ilgili kısıtlar sadece “Having” operatörü ile tanımlanabilmektedir. Diğer bir deyişle “Where” operatörüyle fiziksel veriler üzerinde filtre tanımlanabilirken “Having” operatörü, gruplama fonksiyonları ile hesaplanan sanal veriler üzerinde filtre tanımlanmasında kullanılmaktadır.

Şimdi de “Avg” fonksiyonundan dönen değerler ile filtre uygulamasına bir örnek verelim. Daha önce “Sınav_Id” bilgisi 5 olan sınavdan alınan notların ortalamasını alan SQL kod örneğini yazmıştık ve hesaplanan değer 52 olduğunu görmüştük. Şimdi de sınav not ortalaması 52’den büyük olan sınavların ortalamalarını görüntüleyelim (Şekil 7.18).



“Where” operatörüyle tanımlanan kısıtlar “Having” operatörü içinde de de yer alabilir.



Örnek

•SELECT Sinav_Id, AVG(notu) FROM Sinav_Ogrenci GROUP BY Sinav_Id HAVING AVG(notu) > 52



Fonksiyondan dönen değer için takma ad kullanılmazsa “No column name” olarak görünecektir.

SQLQuery1.sql - Sin...(SINAN\sinan (54))* X

```

SELECT
  Sinav_Id, AVG(notu)
FROM
  Sinav_Ogrenci
GROUP BY
  Sinav_Id
HAVING
  AVG(notu) > 52

```

100 %

Results Messages

	Sinav_Id	(No column name)
1	3	70
2	4	64
3	6	92
4	8	80
5	9	80
6	11	85

M) SINAN\sinan (54) OgrenciBilgi 00:00:00 | 6 rows

Şekil 7.18. Sınav Not Ortalaması 52’den Büyük Olan Sınavların Not Ortalamaları

Bu örnekte gruplama fonksiyonuna takma ad verilmediği için SQL Server tarafından listeleme esnasında sütun adı, “No column name” (kolon adı yok) şeklinde görüntülenmektedir.



Bireysel Etkinlik

- Verilerin gruplanarak sorgulanmasının performans üzerindeki etkilerini araştırınız.
- Öğrencilerin sınavlardan aldıkları notları içeren bir tablo oluşturun ve her bir sınav için 50 ve üzeri not alan öğrenci sayılarını listeleyin.
- 3'ten fazla öğrencinin 50'nin altında not aldığı sınavları listeleyiniz.
- 3'ten fazla sınavdan 50'nin altında not alan öğrencileri listeleyiniz.



Özet

- Bir grup veri, belli bir özelliğe göre istatistiksel analiz için gruplandırılabilir. Bir veya birden fazla tablodan sorgulanan kayıtlar "Group By" deyimini kullanılarak gruplanabilmektedir.
- "Group By" deyimini "Select" yapısı içinde "Where" operatöründen sonra ve "Order By" operatöründen önce gelmektedir.
- Gruplama işlemi yapılırken görüntülenmek istenen alanlar, gruplama işlemine tabi tutulmalıdır. Yani "Group By" deyiminden sonra görüntülenmek istenen alanlar listelenmelidir.
- Gruplanan veriler üzerinde grup fonksiyonları kullanılabilir.
- Grup fonksiyonu kullanılmadan gruplama yapıldığında kayıtlar benzersiz olarak listelenmektedir.
- Gruplama yapılırken select listesi içinde yer alan tüm sütun adları, gruplamaya dâhil edilmelidir.
- Gruplama fonksiyonları bir sütundaki verileri işleme tabi tutarak tek bir değer döndürmektedir.
- Başlıca grup fonksiyonları: MAX, MIN, AVG, COUNT ve SUM.
- Max fonksiyonu ilgili sütundaki en büyük değeri döndürür. Sayısal, metin ve tarih veri türlerindeki sütun adlarını parametre olarak almaktadır. Gruplama yapmadan da gruplama fonksiyonlarının kullanılabilir ve sorgu sonucunda tek kayıt dönmektedir.
- Min fonksiyonu ilgili sütundaki en küçük değeri döndürür. Sayısal, metin ve tarih veri türlerindeki sütun adlarını parametre olarak almaktadır. Yani sütundaki en küçük tarihi, sayıyı veya metni döndürmektedir.
- Avg fonksiyonu ilgili sütun değerlerinin ortalamasını döndürür. Sayısal türde veriler için geçerlidir. Metin ve Tarihsel veri türleri için kullanımı hataya sebep olmaktadır.
- Sum fonksiyonu ilgili sütun değerlerinin toplamını döndürür. Yani parametre olarak aldığı sütunda tutulan verilerin sayısal türde veriler olması gerekmektedir. "Avg" fonksiyonu gibi bu fonksiyon da metin ve tarihsel veri türündeki kolon adlarını, parametre olarak almamaktadır.
- Count fonksiyonu herhangi bir sütunu parametre olarak aldığı anda, ilgili sütun değerlerinin "null" olmadığı kayıt sayısını döndürür.
- Count(*) şeklindeki kullanım ile sorgu sonucu dönen kayıt sayısını verir.
- Birden fazla sütuna göre gruplarken "Group By" operatöründen sonra bu sütun adlarının yazılması gerekmektedir. Diğer taraftan gruplama yapılırken "Select" ifadesi ile görüntülenen bütün sütun adları "Group By" deyimini ile birlikte yazılmalıdır.
- Grup fonksiyonlarının döndürdüğü değerler ile ilgili koşul tanımlamak için "Having" deyimini kullanılmaktadır.
- "Where" operatörüyle tanımlanan kısıtlar "Having" operatörü içinde de yer alabilirken "Having" operatörü içinde tanımlanması gereken kısıtların "Where" operatörü ile kullanılması hataya sebep olmaktadır.
- "Having" deyimini "Group By" deyiminden sonra ve "Order By" deyiminden önce yazılmalıdır.

DEĞERLENDİRME SORULARI

1. Aşağıdaki sorgulardan hangisi personellerin maaşları toplamını getirir?
 - a) SELECT COUNT(Maas) FROM Personel
 - b) SELECT MAX(Maas) FROM Personel
 - c) SELECT TOPLAM(Maas) FROM Personel
 - d) SELECT SUM(Maas) FROM Personel
 - e) SELECT SUM(*) FROM Personel
2. Aşağıdaki sorgulardan hangisi "Birey" tablosundaki "Adi" alanını tekrarsız olarak getirir?
 - a) SELECT Adi FROM Birey GROUP BY Adi, Soyadi
 - b) SELECT Adi FROM Birey GROUP BY Adi
 - c) SELECT Adi FROM Birey GROUP BY Adi HAVING COUNT(*)>1
 - d) SELECT Adi FROM Birey ORDER BY Adi
 - e) SELECT DISTINCT Adi, * FROM Birey
3. Aşağıdaki sorgulardan hangisi "Birey" tablosundaki "Adi" ve "Soyadi" alanlarına göre tekrarlı olan yani adı ve soyadı aynı olan bireyleri listeler?
 - a) SELECT Adi, Soyadi FROM Birey GROUP BY Adi, Soyadi
 - b) SELECT * FROM Birey GROUP BY Adi, Soyadi HAVING COUNT(*)>1
 - c) SELECT Adi, Soyadi FROM Birey GROUP BY Adi, Soyadi HAVING COUNT(*)>1
 - d) SELECT TOP 2 Adi, Soyadi FROM Birey
 - e) SELECT DISTINCT Adi, Soyadi FROM Birey
4. Aşağıdakilerden hangisi en yüksek maaşı alan personelin maaş bilgisini görüntüler?
 - a) SELECT BirimId, MAX(Maas) FROM Personel
 - b) SELECT MAX(Maas) FROM Personel
 - c) SELECT AVG(Maas) FROM Personel
 - d) SELECT Maas FROM Personel GROUP BY MAX(Maas)
 - e) SELECT TOP1 Maas FROM Personel

```
SELECT DogumYeri, COUNT(*) FROM Birey GROUP BY  
DogumYeri HAVING COUNT(*)>11 ORDER BY DogumYeri  
DESC
```

5. Yukarıdaki sorguya göre aşağıdakilerden hangisi yanlıştır?
 - a) Bireyleri doğum yerlerine göre gruplar.
 - b) Doğum yerlerine göre azalan Şekil 7.de sıralar.
 - c) Doğum yeri anı olan birey sayısı 11'den büyük olan doğum yerlerini ve sayılarını görüntüler.
 - d) Doğum yeri bilgisi girilmemiş olan bireyleri hariç tutmaktadır.
 - e) Sorgu sonrası farklı girilmiş doğum yeri bilgisi sayısınca kayıt listeler

6. Aşağıdakilerden hangisi personelleri birimlerine göre gruplayarak her birimde en yüksek maaşı alan personelin maaş bilgisini görüntüler?
- SELECT BirimId, MAX(Maas) FROM Personel GROUP BY BirimId
 - SELECT MAX(Maas) FROM Personel GROUP BY BirimId, Maas
 - SELECT BirimId, AVG(Maas) FROM Personel ORDER BY BirimId
 - SELECT BirimId, Maas FROM Personel GROUP BY MAX(Maas)
 - SELECT BirimId, TOP(Maas) FROM Personel GROUP BY BirimId
7. Aşağıdakilerden hangisi cinsiyet bilgisi girilmeyen (null) birey sayısını verir?
- SELECT COUNT (Cinsiyeti) FROM Birey
 - SELECT COUNT(*) FROM Birey
 - SELECT SUM(*) FROM Birey HAVING Cinsiyeti IS NULL
 - SELECT COUNT(Cinsiyeti) FROM Birey WHERE Cinsiyeti IS NULL
 - SELECT COUNT(*) FROM Birey WHERE Cinsiyeti IS NULL
8. Aşağıdaki sorgulardan hangisi her durumda sıfır değerini döndürmektedir?
- SELECT COUNT(Cinsiyeti) FROM Birey WHERE Cinsiyeti IS NULL
 - SELECT COUNT(*) FROM Birey WHERE Cinsiyeti IS NULL
 - SELECT COUNT(*) FROM Birey
 - SELECT SUM(Cinsiyeti) FROM Birey WHERE 1=2
 - SELECT AVG(Cinsiyeti) FROM Birey
9. Aşağıdaki sorgulardan hangisi her durumda null değerini döndürmektedir?
- SELECT COUNT(Cinsiyeti) FROM Birey WHERE Cinsiyeti IS NULL
 - SELECT COUNT(*) FROM Birey WHERE Cinsiyeti IS NULL
 - SELECT COUNT(*) FROM Birey
 - SELECT SUM(Cinsiyeti) FROM Birey WHERE 1=2
 - SELECT AVG(Cinsiyeti) FROM Birey

“SELECT DogumYeri, Adi, SUM(*) FROM Birey GROUP BY DogumYeri WHERE AVG(Soyadi)>1 ORDER BY DogumYeri DESC”

10. Aşağıdakilerden hangisi verilen sorguda yapılan yanlışlardan biri değildir?
- “Order by” ifadesi cümlede sonunda gelmiş.
 - “*” karakteri “Sum” fonksiyonuna parametre olarak verilmiş.
 - “Avg” fonksiyonu karakter türde bir değeri parametre olarak almış.
 - Gruplama fonksiyonu “Where” ifadesi ile birlikte kullanılmış.
 - “Adi” sütunu “Select” ifadesi içinde yer almış, “Group by” ifadesi içinde yer almamış.

Cevap Anahtarı

1.d, 2.b, 3.c, 4.b, 5d. 6.a, 7.e, 8.a, 9.d

YARARLANILAN KAYNAKLAR

- Adar, İ., 2012. SQL Server 2012 Programlama ve Yönetim. İstanbul, 720 Gözüdeli, Y., 2010. Yazılımcılar için SQL SERVER 2008 R2 ve Veritabanı
Elbahadır, H., 2012. T-SQL SQL SERVER 2012. İstanbul, 294
Programlama. Ankara, 671