GROUP BY Komutu ve Gruplamalı Fonksiyonlar

Bazne elimizdeki verileri önce gruplara ayırıp daha sonra bu gruplardan her birinin;-en büyük, en küçük, toplam, ortalama gibi değerlerini bulmak durumunda kalabiliriz. Bu durumda Gruplamalı fonksiyonlar ve GROUP BY komutu kullanılır.

Bazen verileri gruplandırarak işlem yapmak zorunda kalabiliriz. Genellikle veriler gruplandığında her bir grup için şu işlemlerden biri yapılmak isteniyor demektir; grup toplamı, grup ortalaması, grubun en düşüğü, en fazlası.... Bu bölümde, kitap tablomuzun biraz değişik bir hali üstünde çalışacağız. Bu yapı, normalizasyon kurallarına uymaz, ancak GRUPLAMA konusunu anlamak açısından oldukça yararlı bir yapıdır.

GRUPLAMALI işlemler için kitap tablosu:

Bu türden işlemleri yapan fonksiyonlara GRUPLAMALI FONKSİYONLAR (Aggrigate Functions) denir.
Gruplamalı Fonksiyonlar(Aggregate Functions):
SUM(sutun_adi): İstenilen bir sütundaki değerlerin toplamını verir. Sayı türü olan veriler için geçerlidir.

Örnek-31:
Kitaplığımızda toplam kaç sayfa kitap olduğunu bulmak istersek:

```
SELECT SUM(sayfaSayisi)
FROM Kitap;

>>3611
```

ALANA TAKMA AD(ALIACE): Bazı durumlarda, sorgu sonucunda yer alan bir alanın adını farklı bir ad olarak kullanmak isteyebiliriz. Bu tür durumlarda alan adından sonra AS deyimi gelir ve takma ad belirtilir.

Örnek-32:
Bir önceki örneği alana takma ad ile yaparsak:

```
SELECT SUM(sayfaSayisi) as TOPLAM
FROM Kitap;
```

AVG(sutun_adi):İstenilen bir sütun için ortalama değeri hesaplamada kullanılır. Sayı türü veriler için geçerlidir.

Örnek-33:
Kitaplığımızdaki her bir kitap için ortalama sayfa sayısı (Aritmetik ortalama):

```
SELECT AVG(sayfaSayisi) AS ORTALAMA
FROM Kitap;
```

MAX(sutun_adi): Referans sütuna göre en büyük değeri bulmada kullanılır. Sayı türden veriler için geçerlidir

Örnek-34:
Kitaplığımızdaki en kalın kitap:

```
SELECT MAX(SayfaSayisi)  AS EnKalinKitap
FROM Kitap;
```

MIN(sutun_adi): Referans sütuna göre en küçük değeri bulmada kullanılır. Sayı türden veriler için geçerlidir.

Örnek-35:
Kitaplığımızdaki en ince kitap:

```
SELECT MIN(SayfaSayisi)  AS EnInceKitap
FROM Kitap;
```

Pek kullanılmasa da şu iki fonksiyon da gruplamalı fonksiyondur.
VARIANCE(sutun_adi): Grup varyansını bulur.
STDDEV(grup_adi): Grup standart sapmasını bulur.

COUNT(*): Kayıtların tamamını saymak için kullanılır. Sayı olmasına gerek yoktur.

Örnek-36:
Kitaplığımızdaki Kitapın sayısı:

```
SELECT COUNT(*)
FROM Kitap;

>>16
```

COUNT(sutun_adi): Referans sütuna göre toplam kaç kayıt yer aldığını bulur.

İPUCU:
COUNT(*) ile COUNT(sutun_1) arasındaki fark:
COUNT(*) tüm kayıtların sayısını verirken, COUNT(sutun_1) sadece sutun_1 adlı alan için NULL olmayan kayıtların sayısını verir.

Örnek-37:
Yazarlar tablosunda, tüm yazarların doğum tarihi henüz girilmemiştir. Bu durumda iken, yazarların sayısını COUNT(*) ve bir de COUNT(dogum) olmak üzere iki farklı şekilde seçelim:

```
SELECT COUNT(*)
FROM yazar;

>> 20
```

```
SELECT COUNT(dogum)
FROM yazar;

>> 0
```

COUNT DISTINCT sutun_1 : sutun_1'de kaç farklı değer tekrarladığını sayar.

Örnek-38:
Kütüphanedeki üye sayısı:

```
SELECT COUNT(*)
FROM UYE

>> 9
```

Kütüphanedeki üyelere ait farklı isim sayısı:

```
SELECT COUNT(DISTINCT adi)
FROM uye;
>> 6
```

İPUCU:
GROUP BY'ı ORDER BY ile işlev olarak karıştırmamak gerekir. GROUP BY verilerin gruplara göre sıralanmış olmasından ziyade, gruplara göre hesaplama yapılması ihtiyacı olduğunda kullanılmalıdır. Sonuçların sıralaması ile ilgili her şey için ORDER BY kullanmak gerekir.

Bu kısa örneklerden sonra, grupsal fonksiyonların gruplar üstünde nasıl kullanıldığına yakında bakalım.

Örnek-39:
Elimizdeki Kitapları türüne göre gruplandırdığımızda, her bir grupta kaç kitap yer alıyor?

```
SELECT turu,COUNT(*)

FROM Kitap

GROUP BY turu;

>>

<****************Sonuc34.jpg*****************>
```

Örnek-40:
Elimizdeki Kitapı türlerine göre grupladığımızda her bir türden toplam kaç sayfa kitabımız var?

```
SELECT turu,SUM(sayfaSayisi) AS grupBasiToplamSayfa
FROM Kitap
GROUP BY turu;
```

Örnek-41:
Elimizdeki Kitapları türlerine göre grupladığımızda her bir tür için ortalama kitap sayfa adedini bulalım ama sayfa sayısı 400'den fazla olan Kitapları ve 75'den az olan Kitapları ortalamanın dışında tutalım(Hesaba katmayalım).

```
SELECT turu as TURLER, AVG(SayfaSayisi) AS grupOrtalamaSayfa
FROM Kitap
WHERE sayfaSayisi BETWEEN 75 AND 400
GROUP BY turu;
```

İPUCU:
Kayıt filtrelemede kullanılacak şartlar arasında gruplamalı Fonksiyonu kullanmayı gerektiren bir şart varsa, HAVING deyimi ile belirtilir.

Örnek-42:
Bir arkadaşınız sizin kütüphane olayında otomasyona geçtiğinizi gördü ve şöyle bir soru yöneltti:
Her bir türden toplam kaç kitabın var?
Ve ekledi: ha bu arada, 2 kitaptan az olan türleri saymayacaksın...
İşte cevabı:

```
SELECT turu,COUNT(*)
FROM Kitap
GROUP BY turu
HAVING COUNT(*)>=2;
```

Örnek-43:
Aynı arkadaşınız abarttı ve şu soruyu yöneltti:
Her bir türde toplam kaç sayfa kitabın var?
Ve ekledi
Bu arada, kitap numarası 10'dan küçük olan Kitapın muhtemelen eskidir, sayma. Bir de Ortalaması 175 sayfayı aşmayan türleri de görmek istemiyorum, çünkü muhtemelen bu türlerde yeterince kalın kitaplar yok.

En az sayfa kitap bulunan tür en altta yer alacak şekilde liste verebilir misin?
Öncelikle şu ifadenin sonucuna bakalım:

```
SELECT turu,SUM(sayfaSayisi) sayfaSayisi
FROM Kitap
WHERE kitapNo>10
GROUP BY turu
```

```
SELECT turu,SUM(sayfaSayisi) sayfaSayisi
FROM Kitap
WHERE kitapNo>10
GROUP BY turu
HAVING  AVG(sayfaSayisi)>175
ORDER BY SUM(sayfaSayisi) DESC;
```

HAVING ifadesinden sonra, hikaye türü gelmemektedir. Çünkü grup ortalaması 166'dır ve bu da HAVING şartında verilen değerin altındadır.