

5 dk süre...😊

- En genç çalışanın çalıştığı projelerin numaralarını listeleyiniz.

## Lab-4

tablolarda gruptama, sıralama  
fonksiyonları.

# Grup (Aggregate) Fonksiyonları

- Avg( )
- Count( )
- Min( )
- Max( )
- Sum( )

**SELECT**  
**HAVING**

İfadelerinde kullanılır.

**WHERE**

İfadesinden hemen sonra kullanılmaz.

# Grup (Aggregate) Fonksiyonları

## Sorgu 1:

«Sales» departmanında kaç kişinin çalıştığını, toplam maaşlarını, en yüksek maaşı, en düşük maaşı ve ortalama maaşı bulunuz.

```
SELECT    count(*),  
            sum(salary),  
            max(salary),  
            min(salary),  
            avg(salary)  
FROM department d, employee e  
WHERE     d.dnumber = e.dno AND  
            d.dname = 'Sales'
```

## Sorgu 2:

8 numaralı departmanda çalışan işçilerin ortalama ve toplam maaşlarını bulunuz.

```
SELECT  avg(salary),  
         sum(salary)  
FROM    employee e  
WHERE   e.dno = 8
```

avg(salary)	sum(salary)
40821	571500

# AS anahtar sözcüğü

\* 8 numaralı departmanda çalışan işçilerin ortalama ve toplam maaşlarını bulunuz.

```
SELECT    avg(salary) AS ortalama,  
          sum(salary) AS toplam  
FROM      employee e  
WHERE     e.dno = 8
```

AS kullanmadan  
da isimlendirme  
yapılabilir.

ortalama	toplam
40821	571500

### **Sorgu 3:**

«Middleware» projesinde kaç kişinin çalıştığını ve bu çalışanların ortalama maaşlarını bulunuz.

### Sorgu 3:

«Middleware» projesinde kaç kişinin çalıştığını ve bu çalışanların ortalama maaşlarını bulunuz.

```
SELECT    count(*) AS calisan_sayisi,  
          avg(salary)  
FROM      employee e, works_on w, project p  
WHERE     e.snn = w.essn AND  
          w.pno = p.pnumber AND  
          p.pname = 'Middleware'
```

	calisan_sayisi bigint	avg numeric
1	4	47125.00000000000000



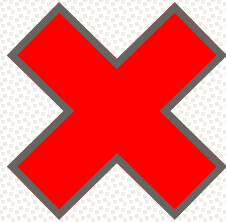
- En genç çalışanın çalıştığı projelerin numaralarını listeleyiniz.

```
SELECT pno
FROM employee, works_on
WHERE ssn=essn AND bdate IN (SELECT MAX(bdate)
                              FROM employee)
```

IN yerine  
= yazılabilir

- En genç çalışanın çalıştığı projelerin numaralarını listeleyiniz.

```
SELECT pno  
FROM employee, works_on  
WHERE ssn=essn AND  
bdate = MAX(bdate)
```



```
Output pane  
Data Output Explain Messages History  
ERROR: aggregate functions are not allowed in WHERE  
LINE 3: WHERE ssn=essn AND bdate = MAX(bdate)  
^  
***** Error *****  
ERROR: aggregate functions are not allowed in WHERE  
SQL state: 42803  
Character: 65
```

# GROUP BY

- Gruplamalar yapıp, grup fonksiyonlarını kullanmayı sağlar.
  - Project tablosunu dnum kolonuna göre gruplandırmak, ve herbir departmanda kaç tane proje olduğunu yazdırmak istersek:

```
SELECT dnum, count(*)  
FROM project  
GROUP BY dnum
```

	dnum smallint	count bigint
1	4	2
2	1	1
3	5	3
4	6	3
5	7	2

# GROUP BY

**Soru :** Bilgisayar Müh. Bölümündeki her bir dersin kaç kişi tarafından alındığını listeleyin.

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM2501	12501105
BLM3561	15011607
EEM4581	12011067
EEM4581	13011705

take tablosu

```
SELECT count(*), code
FROM take
WHERE code LIKE 'BLM%'
GROUP BY code
```

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
BLM3561	15011607

code	id
BLM1551	12501105, 15011607, 12011031
BLM1541	12011031, 12011067
BLM3561	15011607

count(*)	code
3	BLM1551
2	BLM1541
1	BLM3561

sonuç

# ORDER BY

- SELECT ifadesinden döndürülen satırları belirtilen ölçütlere göre artan veya azalan düzende sıralamanıza olanak tanır.

code	id
BLM1551	12501105
BLM1551	<del>15011607</del>
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM2501	12501105
BLM3561	<del>15011607</del>
EEM4581	12011067
EEM4581	<del>12011705</del>

take tablosu

**Soru:** 2012 girişli öğrencileri ve aldıkları dersleri gösterin. Öğrenci numaraları küçükten büyüğe sıralı olsun

```
SELECT code, id
FROM take
WHERE id LIKE '12%'
ORDER BY id
```

code	id
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM4581	12011067
BLM1551	12501105
EEM2501	12501105

sonuç



Artan sıraya göre  
sıralandılar

# ORDER BY

Öğrenci numaraları büyükten küçüğe sıralı olsun istersek **DESC** sözcüğünü kullanırız.

```
SELECT code, id
FROM take
WHERE id LIKE '12%'
ORDER BY id DESC
```

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM2501	12501105
BLM3561	15011607
EEM4581	12011067
EEM4581	13011705

take tablosu

code	id
BLM1551	12501105
EEM2501	12501105
BLM1541	12011067
EEM4581	12011067
BLM1551	12011031
BLM1541	12011031

sonuç

# ORDER BY

```
SELECT  code, id
FROM    take
ORDER BY id DESC, code ASC
veya
ORDER BY id DESC, code
```

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM2501	12501105
BLM3561	15011607
EEM4581	12011067
EEM4581	13011705

take tablosu

code	id
BLM1551	15011607
BLM3561	15011607
EEM4581	13011705
BLM1551	12501105
EEM2501	12501105
BLM1541	12011067
EEM4581	12011067
BLM1541	12011031
BLM1551	12011031

sonuç

#### **Sorgu 4:**

Her bir projede çalışanların ortalama maaşını bulup proje ismine göre alfabetik olarak sıralayınız.



## Sorgu 4:

Her bir projede çalışanların ortalama maaşını bulup proje ismine göre alfabetik olarak sıralayınız.

```
SELECT    pname,  
          avg(salary)  
FROM      employee e, works_on w, project p  
WHERE     e.ssn = w.essn AND  
          w.pno = p.pnumber  
GROUP BY  pname  
ORDER BY  pname
```

	pname character varying (25)	avg numeric
1	Computerization	30000.000000000000
2	DatabaseSystems	54250.000000000000
3	InkjetPrinters	53312.500000000000
4	LaserPrinters	61500.000000000000
5	Middleware	47125.000000000000
6	Newbenefits	31000.000000000000
7	OperatingSystems	42166.666666666667
8	ProductX	27500.000000000000
9	ProductY	31666.666666666667
10	ProductZ	39000.000000000000
11	Reorganization	46000.000000000000

## Sorgu 5:

Her bir departmanda her bir cinsiyetten kaçar işçi olduğunu ve bu işçilerin ortalama maaşlarını bulunuz.

```
SELECT dno,  
       sex,  
       count(*),  
       avg(salary)  
FROM   employee  
GROUP BY dno, sex
```

	dno smallint	sex character (1)	count bigint	avg numeric
1	4	M	1	25000.000000000000
2	8	F	4	37500.000000000000
3	8	M	10	42150.000000000000
4	4	F	2	34000.000000000000
5	5	M	3	36000.000000000000
6	7	M	7	65785.714285714286
7	6	M	7	57285.714285714286
8	5	F	1	25000.000000000000
9	7	F	3	58000.000000000000
10	1	M	1	55000.000000000000
11	6	F	1	79000.000000000000

# HAVING

- Grupları filtrelemek için, WHERE ifadesi yerine **HAVING** ifadesi kullanılır..
- Aggregate fonksiyonlarını WHERE ifadesinden sonra kullanamayız, HAVING ile kullanabiliriz.
- HAVING'den önce GROUP BY ifadesi kullanılmalıdır.
- Örnek : Ortalama maaşın 40000'den fazla olduğu departmanların numaralarını listeleyin.

SELECT dno

FROM employee

GROUP BY dno

HAVING avg(salary)>40000

	dno smallint
1	8
2	1
3	6
4	7

# HAVING

- Soru : 2012 girişli öğrenciler arasında sadece 1 kişi tarafından alınan derslerin kodlarını ve dersi alan öğrencilerin id'lerini listeleyin.

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM2501	12501105
BLM3561	15011607
EEM4581	12011067
EEM4581	13011705

take tablosu

```
SELECT code, id
FROM take
WHERE id LIKE '12%'
GROUP BY code
HAVING count(*)=1
```

code	id
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM4581	12011067
BLM1551	12501105
EEM2501	12501105

code	id
BLM1551	12011031, 12501105
BLM1541	12011031, 12011067
EEM4581	12011067
EEM2501	12501105

code	id
EEM4581	12011067
EEM2501	12501105

sonuç

## Sorgu 6:

5 numaralı departman dışındaki departmanlar arasından, ortalama maaşı 40000\$'dan fazla olan departmanların numaralarını ve bu departmandaki ortalama maaşları bulan sorguyu yazınız.

```
SELECT dno AS departman_no,  
       avg(salary) AS ortalama_maas  
FROM   employee  
GROUP BY dno  
HAVING avg(salary) > 40000  
       AND dno<>5
```

	departman_no smallint	ortalama_maas numeric
1	8	40821.428571428571
2	1	55000.000000000000
3	6	60000.000000000000
4	7	63450.000000000000

Sıra:

SELECT  
FROM  
WHERE  
GROUP BY  
HAVING  
ORDER BY  
LIMIT

# LIMIT

- Sorgu sonucundaki oluşan tabloda **belli bir aralıktaki satırları** almak için kullanılır.

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM2501	12501105
BLM3561	15011607
EEM4581	12011067
EEM4581	13011705

```
SELECT  code, id
FROM    take
LIMIT  3
```

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031

# LIMIT

code	id
BLM1551	12501105
BLM1551	15011607
BLM1551	12011031
BLM1541	12011031
BLM1541	12011067
EEM2501	12501105
BLM3561	15011607
EEM4581	12011067
EEM4581	13011705

```
SELECT  code, id
FROM    take
LIMIT  3 OFFSET 5
```

code	id
EEM2501	12501105
BLM3561	15011607
EEM4581	12011067



# LIMIT

- LIMIT A

- *İlk A tanesi*

- LIMIT A OFFSET B

- *A adet satır B numaralı kayıttan sonra*

## Sorgu 7:

En çok maaşı alan işçinin ismini ve soyismini gösteren sorguyu yazınız. (ORDER BY ve LIMIT kullanarak)

```
SELECT  fname, lname  
FROM    employee  
ORDER BY salary DESC  
LIMIT   1  
        veya  
LIMIT   1 OFFSET 0
```

# İç sorgu ile alternatif çözümü..

```
SELECT      fname, lname  
FROM        employee  
WHERE salary = (SELECT max(salary) FROM employee)
```

# Yöneticisi olmayan kişileri bulunuz

Tablodaki **NULL** değeri, **boş** görünen bir alandaki değerdir.

```
SELECT ssn, superssn  
FROM employee  
WHERE superssn IS NULL
```

Output pane		
Data Output Explain Messages		
	ssn character(9)	superssn character(9)
1	333445555	888665555
2	987654321	888665555
3	888665555	
4	111111100	
5	444444400	
6	555555500	
7	123456789	333445555
8	999887777	987654321
9	666884444	333445555
10	453453453	333445555
11	987987987	987654321
12	111111101	111111100
13	111111102	111111100
14	111111103	111111100
15	222222200	
16	222222201	222222200
17	222222202	222222200
18	222222203	222222200
19	222222204	222222201
20	222222205	222222201
21	333333300	
22	333333301	333333300
23	444444401	444444400
24	444444402	444444400
25	444444403	444444400
26	555555501	555555500
27	666666600	

# Extract anahtar sözcüğü

- **DATE** tipinde belirtilen girdilerden yıl, ay, gün, kaçınıcı hafta vs. bilgisini almanızı sağlar.

```
SELECT extract(YEAR FROM bdate) FROM employee
```

....

CENTURY

DAY

WEEK

MONTH

**SON**