

# Veritabanı Yönetim Sistemleri

---

**Dr. Öğr. Üyesi Ahmet Arif AYDIN**

**SQL-2**

- Relational Calculus
  - Tuple Relational Calculus
  - Domain Relational Calculus
- SQL
  - Select, Create, Update
  - Delete
  - Aggregate Functions(max, min, avg, count,)

# SQL: Structured Query Language

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

```
SELECT * from  
öğrenci WHERE  
ortalama= (select  
min(ortalama) from  
öğrenci) ;
```

```
UPDATE öğrenci  
SET ortalama=ortalama+1  
WHERE  
ortalama =  
(select max(ortalama)  
from öğrenci) ;
```

```
CREATE TABLE öğrenci (  
ögrencino int primary key,  
isim varchar ,  
kullanıcıadı varchar ,  
yaş int,  
ortalama real  
);
```

- INSERT INTO
- SELECT \* veya kolon(lar)
- DISTINCT
- ORDER
- DROP TABLE

# SQL: hstore and json

---

**JSON** : PostgreSQL 9.2 versiyonundan itibaren kullanılmaktadır. JSON tipinde very içeren alan JSON formatında değer gönderir. Kaydedilecek JSON'ın içerdigi alanlar farklı olabilir ve string olarak kaydedilir. Her bir alana direct erişim sağlanır. İki tip kullanılır JSON ve JSONB

```
'{"customer": "John Doe", "items": {"product": "water", "qty": 6}}'
```

**Hstore** : Tek bir deger içerisinde anahtar deger ikililerini text string olarak saklar. Surekli kullanılmayan değerlerin saklanması açısından faydalıdır. Yeni alan eklenşp silinmesini saglar . Performans açısından tavsiye edilir.

```
CREATE EXTENSION hstore;
```

# SQL: UUID

---

Universal Unique Identifier: 128 bit uzunlugunda uluslararası unique olan bir tipdir.

<https://tools.ietf.org/html/rfc4122>

**40e6215d-b5c6-4896-987c-f30f3678f608  
6ecd8c99-4036-403d-bf84-cf8400f67836  
3f333df6-90a4-4fda-8dd3-9485d27cee36**

32 digit içeren hexadecimal

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";  
  
CREATE TABLE contacts (  
    contact_id uuid DEFAULT uuid_generate_v4 (),  
    first_name VARCHAR NOT NULL,  
    last_name VARCHAR NOT NULL,  
    email VARCHAR NOT NULL,  
    phone VARCHAR,  
    PRIMARY KEY (contact_id)  
);
```

# SQL Operatörleri

Sorgularımızı oluştururken SQL operatorları kullanılır

Mantıksal (logical) operatörler  
(AND, OR, and NOT)

a	b	a AND b	a OR b
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	NULL	NULL	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	NULL	FALSE	NULL
NULL	NULL	NULL	NULL

a	NOT a
TRUE	FALSE
FALSE	TRUE
NULL	NULL

Karşılaştırma operatörleri

Operator	Description
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
=	equal
<> or !=	not equal

# SQL Operatörleri

Operator	Description	Example	Result
+	addition	2 + 3	5
-	subtraction	2 - 3	-1
*	multiplication	2 * 3	6
/	division (integer division truncates the result)	4 / 2	2
%	modulo (remainder)	5 % 4	1
^	exponentiation	2.0 ^ 3.0	8
/	square root	/ 25.0	5
/	cube root	/ 27.0	3
!	factorial	5 !	120
!!	factorial (prefix operator)	!! 5	120
@	absolute value	@ -5.0	5
&	bitwise AND	91 & 15	11
	bitwise OR	32   3	35
#	bitwise XOR	17 # 5	20
~	bitwise NOT	~1	-2
<<	bitwise shift left	1 << 4	16
>>	bitwise shift right	8 >> 2	2

Matematiksel  
operatörler

# SQL: Create table and auto-increment

```
CREATE TABLE fruits(  
    id KISITLAMA PRIMARY KEY,  
    name VARCHAR NOT NULL  
);
```



SMALLSERIAL	2 bytes	1 to 32,767
SERIAL	4 bytes	1 to 2,147,483,647
BIGSERIAL	8 bytes	1 to 922,337,2036,854,775,807

```
INSERT INTO fruits(name) VALUES('orange');  
  
INSERT INTO fruits(id,name) VALUES(DEFAULT,'apple');
```

# SQL: Create table and auto-increment

```
create table t2 (id serial primary key, name varchar(20), number integer);  
insert into t2(id, name, number) values(DEFAULT, 'kamil', 75);  
select * from t2;
```

	<b>id</b> integer	<b>name</b> character varying (20)	<b>number</b> integer
1	1	ismail	95
2	2	ali	95
3	3	zeki	95
4	4	kamil	75

# SQL: Create table and auto-increment

```
delete from t2 where name='ali';
```

```
select * from t2;
```

	<b>id</b> integer	<b>name</b> character varying (20)	<b>number</b> integer
1	1	ismail	95
2	3	zeki	95
3	4	kamil	75

# SQL: Create table and auto-increment

```
delete from t2 where name='ali';
```

```
select * from t2;
```

	<b>id</b> integer	<b>name</b> character varying (20)	<b>number</b> integer
1	1	ismail	95
2	3	zeki	95
3	4	kamil	75

Insert işlemi gerçekleştirildiğinde automatik verilen id silinen değerleri kullanmaz.

	<b>id</b> integer	<b>name</b> character varying (20)	<b>number</b> integer
1	1	ismail	95
2	3	zeki	95
3	4	kamil	75
4	5	murat	134

## öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

## dersler

ögrencino	ders.kodu	ders.kredi	ders.not

Birden fazla tabloyu kullanan sorgular oluşturmamız için **dersler** tablosunu oluşturalım

# SQL: Create table and foreign key

---

```
CREATE TABLE dersler (
    ögrencino integer references öğrenci(ögrencino),
    ders.kodu varchar ,
    ders.kredi integer,
    ders.not integer
);
```

dersler

ögrencino	ders.kodu	ders.kredi	ders.not

# SQL: Insert

```
insert into dersler values(17532,'BL44',4,85);
insert into dersler values(17327,'MK25',3,73);
insert into dersler values(17532,'BL13',3,94);
insert into dersler values(17236,'T121',2,45);
insert into dersler values(16458,'KM32',1,52);
insert into dersler values(17236,'BL13',3,53);
insert into dersler values(17347,'BL44',4,75);
insert into dersler values(17532,'TR4',2,65);
```

dersler

öğrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# Tablolar

---

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

dersler

ögrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# Querying (Sorgulama)

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

dersler

ögrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

```
SELECT * FROM dersler, öğrenci  
WHERE öğrenci.ögrencino =  
dersler.ögrencino;
```

# Querying (Sorgulama)

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

dersler

ögrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

```
SELECT * FROM dersler, öğrenci
where öğrenci.ogrencino =
dersler.ogrencino;
```

	ogrencino integer	ders.kodu character varying	ders.kredi integer	ders.not integer	ogrencino integer	isim character varying	kullanıcıadı character varying	yas integer	ortalama real
1	17532	BL44	4	85	17532	Eymen	Eymen234	19	3.2
2	17327	MK25	3	73	17327	Mustafa	m.4417	18	5
3	17532	BL13	3	94	17532	Eymen	Eymen234	19	3.2
4	17236	T121	2	45	17236	Cemil	Cm3418	18	2.9
5	16458	KM32	1	52	16458	Hayri	Hayri95	19	3.8
6	17236	BL13	3	53	17236	Cemil	Cm3418	18	2.9
7	17347	BL44	4	75	17347	Kemal	Kml45	17	3.7
8	17532	TR4	2	65	17532	Eymen	Eymen234	19	3.2

# SQL: Querying (Sorgulama)

öğrenci

öğrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

dersler

öğrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

Eymen öğrencisinin almış  
olduğu bütün dersleri  
listeleyen sql sorgusunu  
yazınız.

?

# SQL: Querying (Sorgulama)

öğrenci

ogrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

dersler

ogrencino	ders. kodu	ders. kredi	ders. not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

Eymen öğrencisinin almış  
olduğu bütün dersleri  
listeleyen sql sorgusunu  
yazınız.

```
SELECT * FROM dersler d where d.ogrencino =  
(select ogrencino from öğrenci o where o.isim='Eymen');
```

# SQL: Querying (Sorgulama)

öğrenci

ogrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

Eymen öğrencisinin almış olduğu bütün dersleri listeleyen sql sorgusunu yazınız.

dersler

ogrencino	ders. kodu	ders. kredi	ders. not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

SELECT \* FROM dersler d where d.ogrencino =  
(select ogrencino from öğrenci o where o.isim='Eymen') ;

	ogrencino integer	ders.kodu character varying	ders.kredi integer	ders.not integer
1	17532	BL44	4	85
2	17532	BL13	3	94
3	17532	TR4	2	65

# SQL: Querying (Sorgulama)

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

dersler

ögrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

Kemalin aldığı derslerden kredisi 4 den  
küçük olanları listeleyiniz

?

# SQL: Querying (Sorgulama)

öğrenci

ogrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

dersler

ogrencino	ders. kodu	ders. kredi	ders. not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

Kemalin aldığı derslerden kredisi 4 den  
küçük olanları listeleyiniz

```
SELECT * FROM dersler d
where d.ogrencino =
( select ogrencino from öğrenci o
  where o.isim='Kemal')
AND d.ders.kredi<4 ;
```

# SQL: LIKE Operatörü

Like operatörü belirtilen **pattern** (örüntü) e uyan sonuçları listelemeyi sağlar.  
LIKE operatörü case sensitivedir. ILIKE case-sensitive değildir.

*string* **LIKE** *pattern*

~~      LIKE.      !~~      NOT LIKE

*string* **NOT LIKE** *pattern*

~~\*    ILIKE      !~~\* NOT ILIKE

*string* **ILIKE** *pattern*

—

(alt tire)  
(tek karakter)

%

(0-n arası)  
any number of characters)

# SQL: LIKE Operatörü

```
SELECT * FROM dersler where  
dersler."ders.kodu" LIKE '_ _ _ 4' ;
```

dersler			
öğrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# SQL: LIKE Operatörü

```
SELECT * FROM dersler where  
dersler."ders.kodu" LIKE '_ _ _ 4' ;
```

	ogrencino integer	ders.kodu character varying	ders.kredi integer	ders.not integer
1	17532	BL44	4	85
2	17347	BL44	4	75

dersler			
ögrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# SQL: LIKE Operatörü

```
SELECT * FROM dersler where  
dersler."ders. kodu" LIKE 'BL%' ;
```

dersler			
öğrencino	ders. kodu	ders. kredi	ders. not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# SQL: LIKE Operatörü

```
SELECT * FROM dersler where  
dersler."ders.kodu" LIKE 'BL%' ;
```

	ogrencino	ders.kodu	ders.kredi	ders.not
	integer	character varying	integer	integer
1	17532	BL44	4	85
2	17532	BL13	3	94
3	17236	BL13	3	53
4	17347	BL44	4	75

dersler			
ögrencino	ders. kodu	ders. kredi	ders. not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# SQL: As

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

**As** ifadesi geçici bir isim Tanımlamaya yarar.

```
select avg (ortalama) AS "sınıfortalaması" from öğrenci;
```

	sınıfortalaması
	double precision
1	3.72000002861023

# SQL: Alias (Değişken) Tanımlama

---

**SELECT sutunadı AS yeni\_isim FROM table;**

```
SELECT * FROM  
    dersler d, öğrenci o  
where  
    d.ogrencino = o.ogrencino;
```

AS  
kullanılmayabilir (optional)

# SQL: Limit

`SELECT * FROM dersler limit 4;`

ilk 4 satiri listeler

dersler			
ögrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# SQL: Limit

SELECT \* FROM dersler limit 4;

ilk 4 satiri listeler

	ogrencino integer	ders.kodu character varying	ders.kredi integer	ders.not integer
1	17532	BL44	4	85
2	17327	MK25	3	73
3	17532	BL13	3	94
4	17236	T121	2	45

dersler

ögrencino	ders.kodu	ders.kredi	ders.not
17532	BL44	4	85
17327	MK25	3	73
17532	BL13	3	94
17236	T121	2	45
16458	KM32	1	52
17236	BL13	3	53
17347	BL44	4	75
17532	TR4	2	65

# SQL: Between

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

Aralık belirlemede  
kullanılır

```
select * from öğrenci where yaşı between 16 and 18;
```

# SQL: Between

öğrenci

ogrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

Aralık belirlemede  
kullanılır

select \* from öğrenci where yaş between 16 and 18;

	ogrencino integer	isim character varying	kullanıcıadı character varying	yas integer	ortalama real
1	17347	Kemal	Kml45	17	3.7
2	17236	Cemil	Cm3418	18	2.9
3	17327	Mustafa	m.4417	18	5

# SQL: IN

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

Verilen liste içinde  
arama yapar

Select \* from öğrenci where yaş IN (18,19);

# SQL: IN

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

Verilen liste içinde  
arama yapar

Select \* from öğrenci where yaş IN (18,19);

	ogrencino integer	isim character varying	kullanıcıadı character varying	yas integer	ortalama real
1	17532	Eymen	Eymen234	19	3.2
2	17236	Cemil	Cm3418	18	2.9
3	16458	Hayri	Hayri95	19	3.8
4	17327	Mustafa	m.4417	18	5

# SQL: IN

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

IN komutunu kullanarak  
Eymen ve Cemilin  
bütün bilgilerini listeleyiniz

Select \* from öğrenci where isim IN ('Eymen', 'Cemil');

# SQL: FETCH

öğrenci

ögrencino	isim	kullanıcıadı	yaş	ortalama
17532	Eymen	Eymen234	19	3.2
17327	Mustafa	m.4417	18	4
17347	Kemal	Kml45	17	3.7
17236	Cemil	Cm3418	18	2.9
16458	Hayri	Hayri95	19	3.8

```
SELECT
    öğrencino,
    isim
FROM
    öğrenci
ORDER BY
    isim
OFFSET 2 ROWS
FETCH FIRST 1 ROW ONLY;
```

Istenilen sıradan başlamak kaydıyla istelenen sayıda satırı listelemeyi sağlar

# SQL: GROUP BY

**SELECT kolon listesi FROM tablo ismi WHERE [ şartlar] GROUP BY kolonlar;**

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

[https://www.w3schools.com/sql/sql\\_groupby.asp](https://www.w3schools.com/sql/sql_groupby.asp)

# SQL: GROUP BY

---

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
ORDER BY COUNT(CustomerID) DESC;
```

COUNT(CustomerID)	Country
13	USA
11	France
11	Germany
9	Brazil
7	UK
5	Mexico
5	Spain
4	Venezuela
3	Argentina

# SQL: Having

```
SELECT COUNT(CustomerID), Country  
FROM Customers  
GROUP BY Country  
HAVING COUNT(CustomerID) > 5  
ORDER BY COUNT(CustomerID) DESC;
```

COUNT(CustomerID)	Country
13	USA
11	France
11	Germany
9	Brazil
7	UK