

# Veritabanı Yönetim Sistemleri

---

Dr. Öğr. Üyesi Ahmet Arif AYDIN

Relatioal Calculus (İlişkisel Hesap)  
SQL (Giriş)

- İlişkisel veri modeli (relational data model)
  - Şema, Tablo
  - Bütünlük Kısıtlamaları (Primary Key, Foreign Key )
  - ER Model 'den İlişkisel Modele Geçiş
- İlişkisel Cebir (relational algebra)
  - Seçim, İzdüşüm, Fark, Birleşim
  - Kartezyen Çarpımı, Bölme, Bitiştirme

- VTYS'lerin sorgulama işleminin temelinde mantık(logic) bulunmaktadır.
- Sorguları ve kısıtlamaları oluşturmak için mantıksal ifadeler kullanılır
  - Logical Expressions - Relational Calculus Expressions
- İlişkisel veri tabanları küme tabanlıdır
  - Entity set
  - Relationship set

ifade 1  $\equiv$  ifade 2

ifade 1 denktir ifade 2

ifade1 ve ifade2 birebir aynı ise (identically equal)  
ikisinin anlamları da aynıdır. Yani bir işlemde ifade 1  
yerine ifade 2 yazılabilir.

Implication : gerektirmek, ima etmek

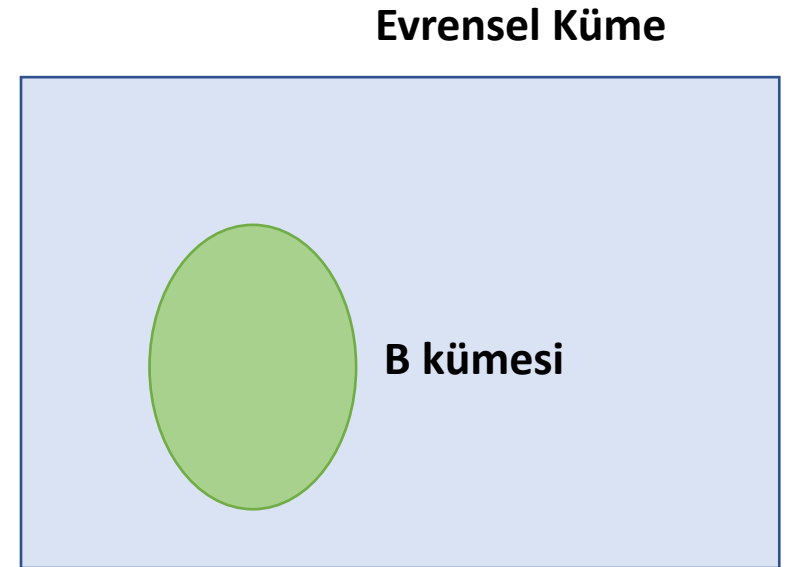
$$p \rightarrow q$$

- If  $p$  then  $q \equiv (\text{not } p) \text{ or } q$
- Eğer  $p$  doğru ise  $q$  da doğrudur (tersi doğru değildir)
- $p$  (hava kapalıdır)  $\rightarrow$   $Q$  (güneş görünmemektedir)
- $\text{not } p \rightarrow \text{not } Q$

$$p \rightarrow q \text{ eşit değildir } q \rightarrow p$$

Bir ifadenin değilinin değil  
(olumsuzun olumsuzu) kendisidir.

$$\text{not}(\text{not } p) \equiv p$$



## DeMorgan's Law

- $\text{not } (p \text{ and } q) \equiv (\text{not } p) \text{ or } (\text{not } q)$
- $\text{not } (p \text{ or } q) \equiv (\text{not } p) \text{ and } (\text{not } q)$

- $p \text{ and } (q \text{ or } r) \equiv (p \text{ and } q) \text{ or } (p \text{ and } r)$
- $p \text{ or } (q \text{ and } r) \equiv (p \text{ or } q) \text{ and } (p \text{ or } r)$



- ❑ İlişkisel cebirin (relational algebra) bir alternatifidir.
- ❑ Declerative'dir (cevapların nasıl hesaplanacağını detayları verilmez)
- ❑ *SQL dilinin ortaya çıkmasında büyük etkisi olmuştur.*

- ❑ Demet İlişkisel Hesap (Tuple Relational Calculus (TRC))
- ❑ Alan ilişkisel Hesap (Domain Relational Calculus (DRC )

$\{ t \mid \text{şart}(t) \}$  şart alanı aşağıdaki ifadelerden oluşabilir:

- atomik formül
- karşılaştırma operatörleri ( $.$ ,  $<$ ,  $\leq$ ,  $=$ ,  $\neq$ ,  $>$ ,  $\geq$ )
- $\neg P = \underline{\text{not}} P$  ( $P$  nin değili)
- $P \wedge Q = P \underline{\text{ve}} Q$
- $(P \vee Q) = P \underline{\text{veya}} Q$
- $P \rightarrow Q = P \text{ doğru ise } Q \text{ da doğrudur}$
- $\exists R ( \text{şart} (R) )$ : ( $\exists$ : en az bir tane var( *there exists* ))
- $\forall K ( \text{şart} (K) )$  ( $\forall$ : hepsi için (for all))

$\{ t \mid P(t) \}$  veya  $\{ t \mid \text{şart}(t) \}$

- $t$  = şartı sağlayan tuple'lar (tabloda bulunan satırlar)
- $\{ t \mid t \in \text{işçi} \wedge t[\text{yaş}] > 40 \}$ 
  - Yaşı 40 dan büyük olan işçilerin bütün bilgilerini (tuple) getir
- $\{ t \mid \text{personel}(t) \text{ and } t.\text{bölümno} = 10 \}$ 
  - 10 nolu bölümde çalışan personelin hepsinin bilgilerini (tuple) getir

$$\{ t \mid P(t) \} \text{ veya } \{ t \mid \text{şart}(t) \}$$

F Blok' da bulunan tüm *akademik personel* in bilgilerini bulunuz

$$\begin{aligned} \{ t \mid & \exists s \in \text{akademikpersonel} ( t[\text{isim}] = s[\text{isim}] \\ & \wedge \exists u \in \text{bölüm} ( \\ & \quad u[\text{bölümadi}] = s[\text{bölümadi}] \\ & \quad \wedge u[\text{bina}] = \text{"F Blok"} ) ) \\ & \} \end{aligned}$$

## KullanıcıReyting

kullanıcıno	isbn	reyting
276746	0553561618	0
276746	055356451X	2
276746	0451166892	3
276746	0786014512	0
276747	0060517794	9
276747	0451192001	0
276751	3596218098	8
276754	0684867621	8
276755	0451166892	5

University of California Irvine

<http://archive.ics.uci.edu/ml/index.php>

278,858 kullanıcı  
1,149,780 reyting  
271,379 kitaplar

Reytingi 0 olan kitapları bul

$\{ t \mid \text{şart}(t) \}$

$\{ t \mid k \in \text{KullanıcıReyting}$

$(t[\text{reyting}] = k[\text{reyting}] \wedge k[\text{reyting}] = 0)$

$\}$



kullanıcıno	isbn	reyting
276746	0553561618	0
276746	055356451X	2
276746	0451166892	3
276746	0786014512	0
276747	0060517794	9
276747	0451192001	0
276751	3596218098	8
276754	0684867621	8
276755	0451166892	5

$\{ \langle \text{sütun1}, \text{sütun2}, \text{sütun3}, \dots \text{sütun } n \rangle \mid \rho(\text{sütun 1}, \text{sütun 2}, \text{sütun 3}, \dots \text{sütun } n) \}$

Hesaplanan sonuç: tuple (satır) yerine seçilen sütunlardır.

Yaşı 30 dan fazla olan akademik personelin bilgilerini yazdırınız

$\{ \langle id, isim, bölüm, yaş \rangle \mid \langle id, isim, bölüm, yaş \rangle \in akademikpersonel \wedge yaş > 30 \}$

Yaşı 30 dan fazla olan akademik personelin sadece id bilgisini yazdırınız

$\{ \langle id \rangle \mid \langle id, isim, bölüm, yaş \rangle \in akademikpersonel \wedge yaş > 30 \}$



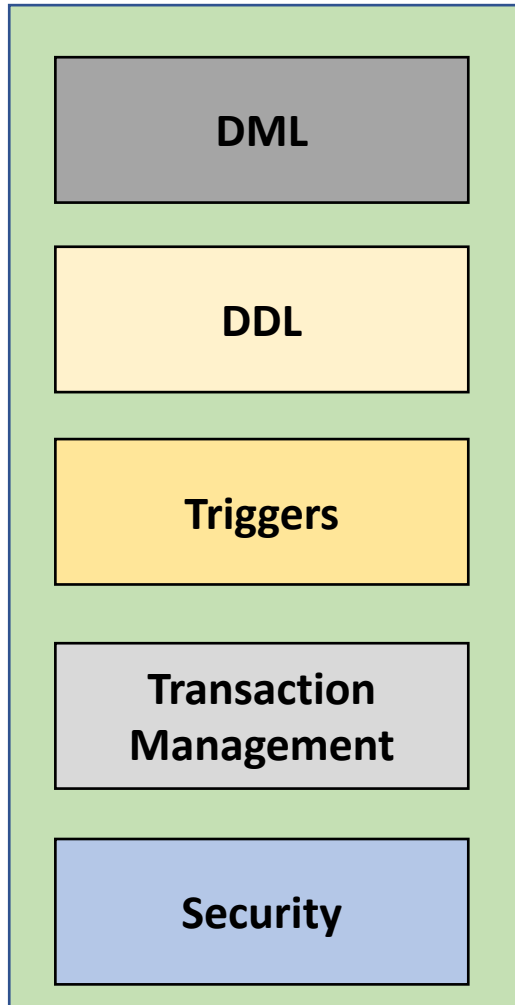
Reytingi 5 den büyük olan kitapların  
isbn ve reyting alanlarını getir.

$\{ \langle isbn, reyting \rangle \mid$

$\langle kullanıcıid, isbn, reyting \rangle \in$   
 $kullanıcıreyting \wedge reyting > 5 \}$

kullanıcıno	isbn	reyting
276746	0553561618	0
276746	055356451X	2
276746	0451166892	3
276746	0786014512	0
276747	0060517794	9
276747	0451192001	0
276751	3596218098	8
276754	0684867621	8
276755	0451166892	5

SQL: 1974-1980 arasında IBM tarafından geliştirilen ve RDBMS' lerde kullanılan bir standarttır



**Data Manipulation Language:** SQL'in satır ekle, sil , değiştir komutlarını içerir.

**Data Definition Language:** SQL tablolarının ve görünüşlerin oluşturulması, silinmesi, ve değiştirilmesi komutlarını içerir.

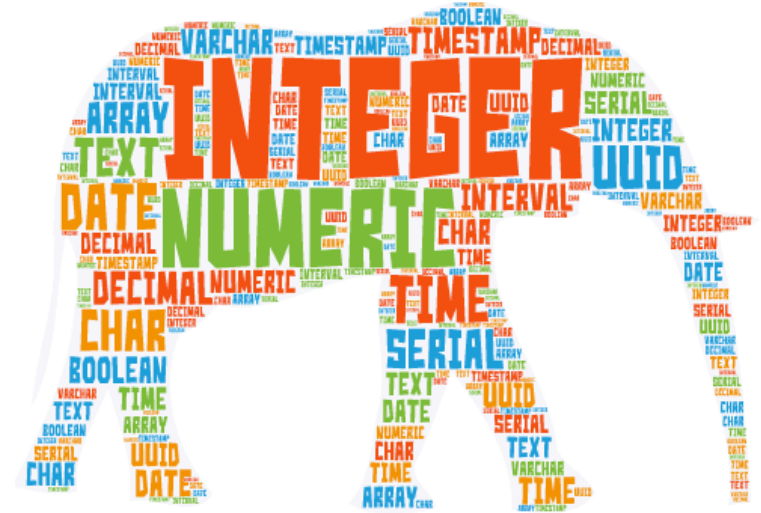
**Triggers:** Veritabanında yapılan bazı değişikliklerin yapılmasıyla tetiklenen işlemlerdir (action)

**Transaction Management:** Hareket yönetimi ile alakalı tanımlamaların yapıldığı kısımdır.

**Security:** SQL yardımıyla kullanıcıların veritabanı nesnelere erişim denetimi sağlanır.

- ❑ SQL sorgularında spaces, tabs, and newlines dikkate alınmaz
- ❑ -- komut satırı ekler
- ❑ SQL case sensitive değildir.
  - SELECT
  - Select
  - select

- ❑ Boolean (True-False)
- ❑ Karakter Tipi (char, varchar, text)
- ❑ Sayısal (integer, float(n), numeric (tam, ondalıklı))
- ❑ Zaman ile alakalı
  - ❑ Date (yyyy-mm-dd)
  - ❑ Time (HH:MM:SS)
  - ❑ Timestamp (2018-11-05 13:35:06.980471+03)
  - ❑ Interval (6 years 5 months 4 days 3 hours 2 minutes 1 second)



Değer	Char (4)	Ayrılan Alan
''	' '	4 bytes
'ab'	'ab '	4 bytes
'abcd'	'abcd'	4 bytes
'abcdefgh'	'abcd'	4 bytes

char (n)  
Ayrılan alan sabit

Varchar(4)	Ayrılan Alan
''	1 byte
'ab'	3 bytes
'abcd'	5 bytes
'abcd'	5 bytes

varchar (n)  
Ayrılan alan değişken

<code>character varying(n), varchar(n)</code>	variable-length with limit
<code>character(n), char(n)</code>	fixed-length, blank padded
<code>text</code>	variable unlimited length

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

```
CREATE TABLE öğrenci (  
    öğrencino int primary key,  
    isim varchar(20) ,  
    kullanıcıadı varchar(20) ,  
    yaş int,  
    ortalama real  
);
```

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

**DROP TABLE öğrenci;**

öğrencino	isim	kullaniciadi	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

```
insert into öğrenci values (17532, 'Eymen', 'Eymen234',19,3.2);
```

```
insert into öğrenci (öğrencino, isim, kullaniciadi, yas, ortalama)  
values (17532, 'Eymen', 'Eymen234',19,3.2);
```



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

Tabloda bulunan  
veriyi listeler

Select \* FROM 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

**select** öğrencino, isim **from** öğrenci;

projektion

	ogrencino integer	isim character varying
1	17532	Eymen
2	17327	Mustafa
3	17347	Kemal
4	17236	Cemil
5	16458	Hayri

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

```
select * from öğrenci  
where yas>17 and  
ortalama>3.0;
```

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

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

```
select * from öğrenci  
order by yas ASC;
```

	ogrencino integer	isim character varying	kullanıcıadı character varying	yas integer	ortalama real
1	17347	Kemal	Kml45	17	3.7
2	17327	Mustafa	m.4417	18	4
3	17236	Cemil	Cm3418	18	2.9
4	17532	Eymen	Eymen234	19	3.2
5	16458	Hayri	Hayri95	19	3.8

ASC (Artan)  
DESC (Azalan)

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

```
select distinct yas from  
öğrenci order by yas;
```

	yas integer
1	17
2	18
3	19

Birden fazla satırdan sonuç olarak **sadece bir satır** üreten fonksiyonlardır

- ❑ **count** (eleman sayısı)
- ❑ **sum** (toplama işlemi)
- ❑ **avg** (average) ortalama
- ❑ **max** (maximum)
- ❑ **min** (minimum)

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

```
select avg (ortalama)
from öğrenci ;
```

	avg double precision
1	3.52000002861023

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

```
select * from öğrenci
where ortalama =(select
max (ortalama) from
öğrenci) ;
```

	ogrencino integer	isim character varying	kullanıcıadı character varying	yas integer	ortalama real
1	17327	Mustafa	m.4417	18	4



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

```
select * from öğrenci
where ortalama =(select
min (ortalama) from
öğrenci) ;
```

	ogrencino integer	isim character varying	kullanıcıadı character varying	yas integer	ortalama real
1	17236	Cemil	Cm3418	18	2.9

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

```

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

	ogrencino integer	isim character varying	kullanıcıadı character varying	yas integer	ortalama real
1	17532	Eymen	Eymen234	19	3.2
2	17347	Kemal	Kml45	17	3.7
3	17236	Cemil	Cm3418	18	2.9
4	16458	Hayri	Hayri95	19	3.8
5	17327	Mustafa	m.4417	18	5

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

X

```
DELETE from öğrenci where ortalama= (select min
(ortalama) from öğrenci) ;
```

Universal Unique Identifier: 128 bit uzunluğunda uluslararası unique olan bir tiptir.

<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-oss";

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)
);
```

```
CREATE TABLE contacts (  
  id serial PRIMARY KEY,  
  name VARCHAR (100),  
  phones TEXT []  
);
```

```
INSERT INTO  
contacts (name, phones)  
VALUES ('John Doe', ARRAY [ '(408) -589-5846' , '(408) -589-5555' ] ) ;
```

```
CREATE TABLE orders (  
ID serial NOT NULL PRIMARY KEY,  
info json NOT NULL  
);
```

```
INSERT INTO orders (info)  
VALUES
```

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

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

- ❑ Sütun içerisinde bir anahtar-> değer yapısı oluşturmayı sağlar.
- ❑ Değerler string olarak saklanır.
- ❑ Sürekli kullanılmayan değerlerin saklanması açısından faydalıdır.
- ❑ Yeni alan eklenip silinmesini sağlar .
- ❑ Performans açısından tavsiye edilir.
- ❑ Hstore un kullanılabilmesi için `CREATE EXTENSION hstore;`

Sütun içerisinde bir anahtar-> değer yapısı oluşturmayı sağlar.

```
CREATE EXTENSION hstore;
```

```
CREATE TABLE books (  
  id serial primary key,  
  title VARCHAR (255),  
  attr hstore  
);
```

```
INSERT INTO books (title, attr) VALUES  
( 'PostgreSQL Tutorial',  
  "paperback" => "243",  
  "publisher" => "postgresqltutorial",  
  "language"  => "English",  
  "ISBN-13"   => "978-1449370000",  
  "weight"    => "11.2 ounces"  
);
```