

# Veritabanı Yönetim Sistemleri

---

Dr. Öğr. Üyesi Ahmet Arif AYDIN

Relatioal Calculus (İlişkisel Hesap)  
SQL-1 (structured query language) (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.
- Veritabanına yönlendirilecek sorgular ve kısıtlamalar mantıksal ifadeler ile oluşturulur.
  - Logical Expressions - Relational Calculus Expressions
- Veritabanları üzerinde gerçekleştirilen işlemler **set teorisine** göre gerçekleştirilir
  - Entity set
  - Relationship set

# Dönüşüm Kuralları (transformation rules)

---

$A$  (ifade 1)  $\equiv$   $B$  (ifade 2)

$A$  denktir  $B$

$A$  ve  $B$

denk ise (identically equal)

Her bir ifadenin anlamları da aynıdır

Dolayısıyla bir işlem gerçekleştirilirken

ifade 1 yerine ifade 2 yazılabilir.

# Implication Law (önerme)

Implication : gerektirmek, ima etmek

$$P \rightarrow Q$$

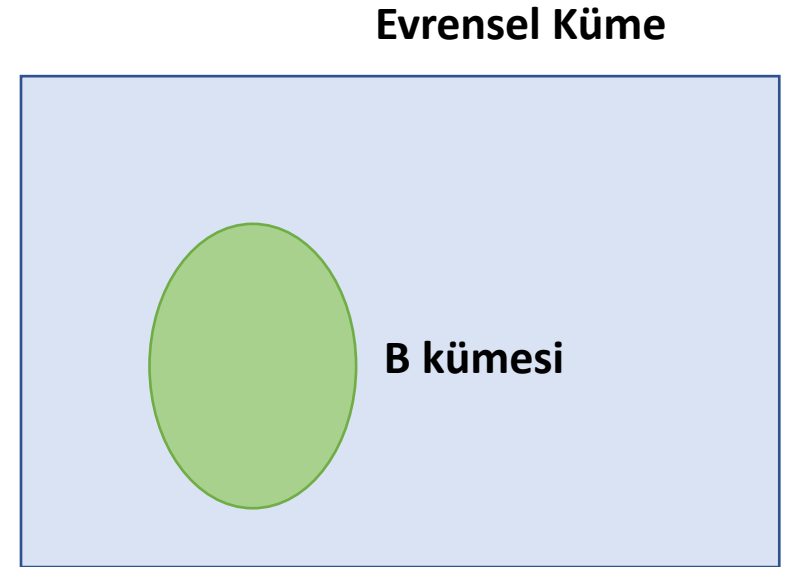
- If p then q  $\equiv$  (not p) or q
- Eğer p doğru ise q da doğrudur (tersi doğru değildir)
- P (hava kapalıdır)  $\rightarrow$  Q (güneş görünmemektedir)
- not P  $\rightarrow$  not Q

$$P \rightarrow Q \text{ eşit değildir } Q \rightarrow P$$

# Double Negation Law

Bir ifadenin değilinin değil  
(olumsuzun olumsuzu) kendisidir.

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



## DeMorgan's Law

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

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

## Dağıtım Kuralı (distributive rules)

---

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

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

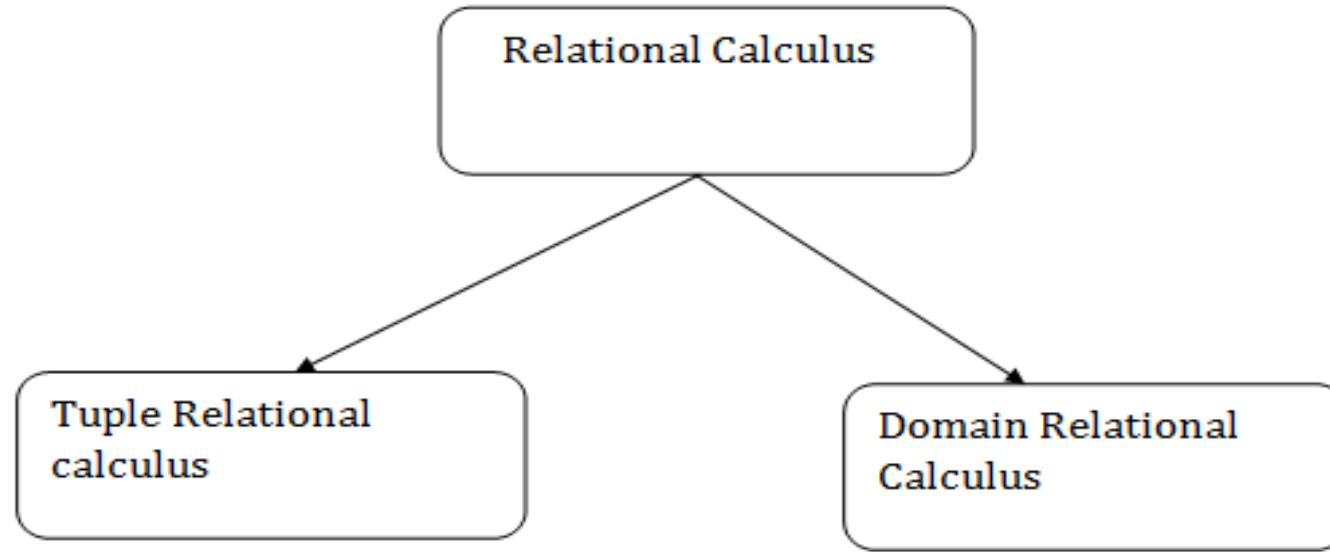


# İlişkisel Hesap (relational calculus)

---

- ilişkisel hesap ilişkisel cebirin (relational algebra) bir alternatifidir.
- Declerative'dir
  - İşlemler için önermeler oluşturulur
  - işlemlerin nasıl gerçekleştirileceği (cevapların nasıl hesaplanacağını) detayları verilmez
- *SQL dilinin ortaya çıkmasında büyük etkisi olmuştur.*

# İlişkisel Hesap (relational calculus)



TRC

Demet İlişkisel Hesap

$\{ t \mid \text{şart}(t) \}$

DRC

Alan ilişkisel Hesap

$\{ \langle \text{sütun1}, \text{sütun2}, \dots, \text{sütun } n \rangle \mid$

$P(\text{sütun 1}, \text{sütun 2}, \dots, \text{sütun } n) \}$

# Demet İlişkisel Hesap: Tuple relational calculus (TRC)

$\{ t \mid \text{şart}(t) \}$  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))

# Demet İlişkisel Hesap: Tuple relational calculus (TRC)

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

- $t$  = bir ilişkide şartı sağlayan tuple (row ) 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

## Demet İlişkisel Hesap: Tuple relational calculus (TRC)

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

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

{ t |

$\exists s \in \text{akademikpersonel} ( t[\text{isim}] = s[\text{isim}] \wedge$

$\exists u \in \text{bölüm} ( u[\text{bölümadi}] = s[\text{bölümadi}] \wedge u[\text{bina}] = \text{"F Blok"} )$

)

}

# Örnek Veritabanı

## 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

$\{ t \mid \text{şart}(t) \}$

{

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

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

}

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

## Alan İlişkisel Hesap: Domain relational calculus (DRC)

---

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

Hesaplanan sonuç: ilişki satırında bulunan sütunlardır (kolon).



## Alan İlişkisel Hesap: Domain relational calculus (DRC)

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

Hesaplanan sonuç: ilişki satırında bulunan sütunlardır (kolon).

Yaşı 30 dan fazla olan akademik personelin bilgilerini yazdırınız

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

## Alan İlişkisel Hesap: Domain relational calculus (DRC)

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

Hesaplanan sonuç: ilişki satırında bulunan sütunlardır (kolon).

Yaşı 30 dan fazla olan akademik personelin id ve yaş bilgilerini listeleyiniz

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

# Alan İlişkisel Hesap: Domain relational calculus (DRC)

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

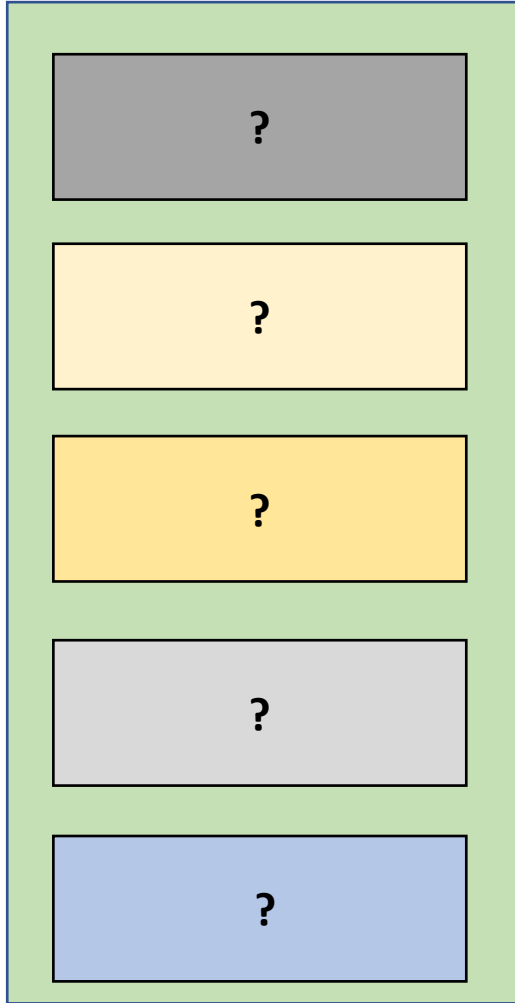
{  
     $\langle isbn, reyting \rangle \mid$   
         $\langle kullanıcıid, isbn, reyting \rangle \in$   
         $kullanıcıreyting \wedge reyting > 5$   
}

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

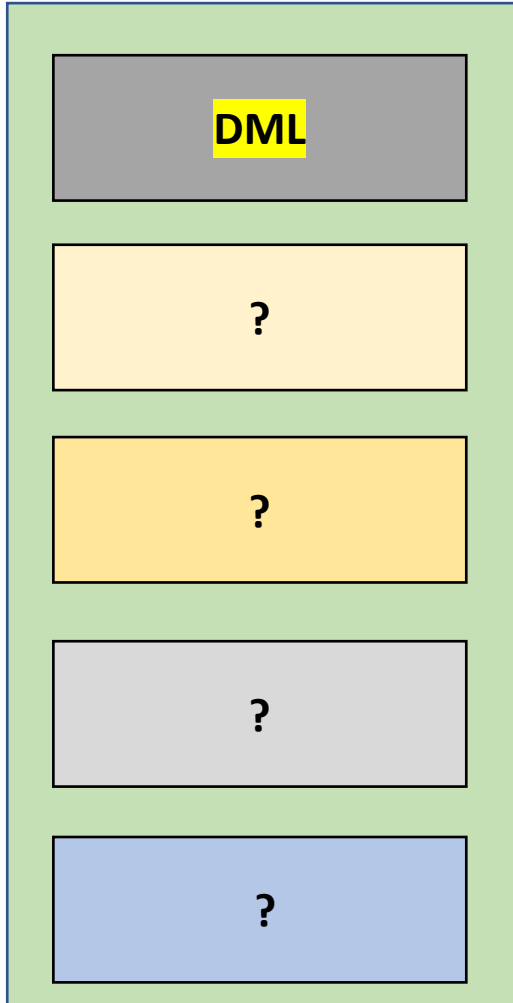
# SQL (Structured Query Language )

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



# SQL (Structured Query Language )

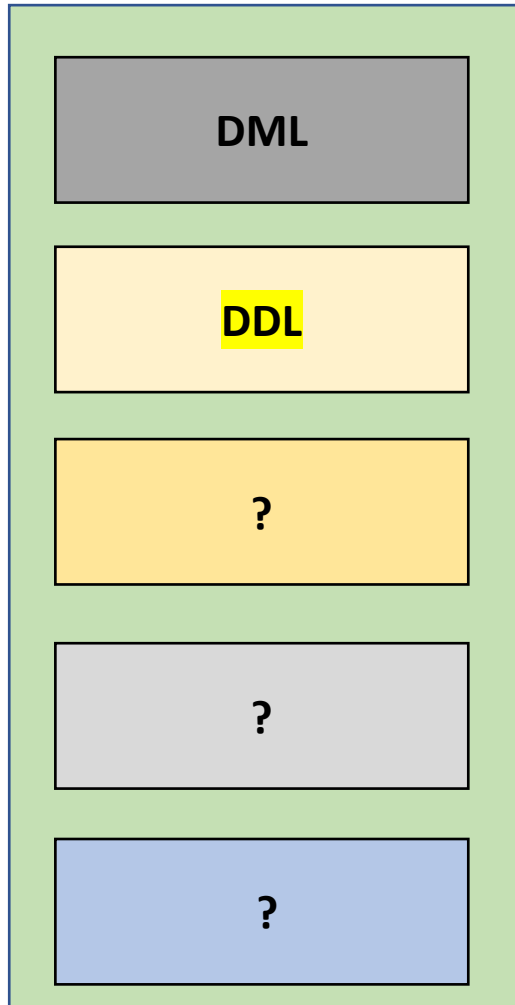
SQL: 1974-1980 arasında IBM tarafından geliştirilen ve RDBMS' lerde kullanılan bir standarttır



**Data Manipulation Language:** SQL'in satır ekle, sil , değiştir komutlarını içerir. (Insert, Update, Delete, Trigger)

# SQL (Structured Query Language )

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

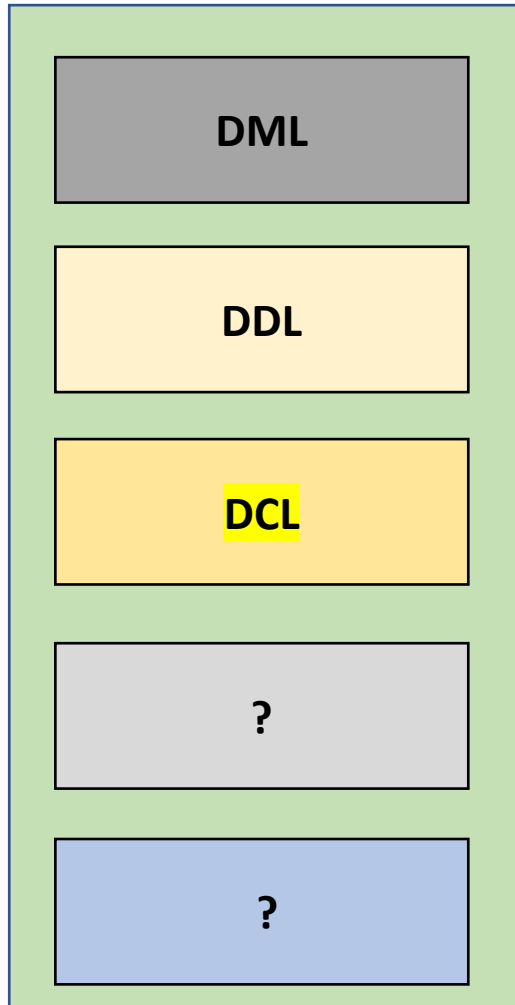


**Data Manipulation Language:** SQL'in satır ekle, sil , değiştir komutlarını içerir. (Insert, Update, Delete, Trigger)

**Data Definition Language:** SQL tablolarının ve görünüşlerin oluşturulması, silinmesi, ve değiştirilmesi komutlarını içerir. (Create, Drop, Alter, Truncate)

# SQL (Structured Query Language )

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



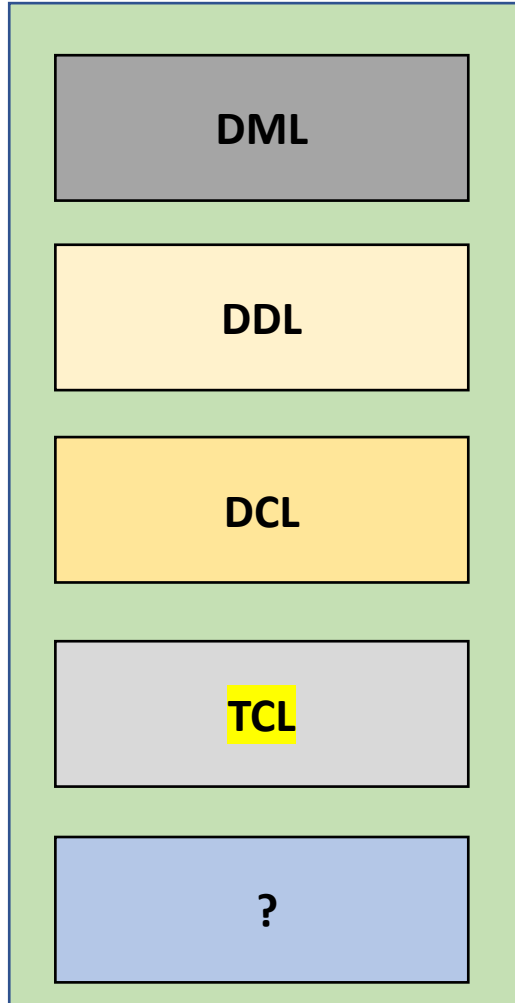
**Data Manipulation Language:** SQL'in satır ekle, sil , değiştir komutlarını içerir. (Insert, Update, Delete, Trigger)

**Data Definition Language:** SQL tablolarının ve görünüşlerin oluşturulması, silinmesi, ve değiştirilmesi komutlarını içerir. (Create, Drop, Alter, Truncate)

**Data Control Language:** Veritabanı kullanıcılarına nesnelere erişim iznin verilmesini veya kısıtlanmasını sağlar. Security (Grant, Revoke)

# SQL (Structured Query Language )

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



**Data Manipulation Language:** SQL'in satır ekle, sil , değiştir komutlarını içerir. (Insert, Update, Delete, Trigger)

**Data Definition Language:** SQL tablolarının ve görünüşlerin oluşturulması, silinmesi, ve değiştirilmesi komutlarını içerir. (Create, Drop, Alter, Truncate)

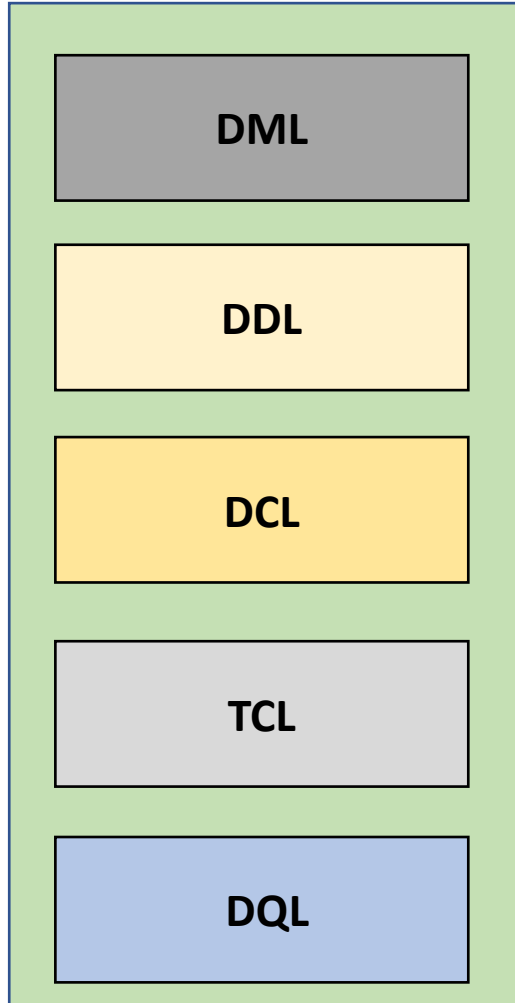
**Data Control Language:** Veritabanı kullanıcılarına nesnelere erişim iznin verilmesini veya kısıtlanmasını sağlar . Security (Grant, Revoke)

**Transaction Control Language:** Hareket Yönetimi ile alakalı olan kısım (Commit, Rollback, Savepoint)



# SQL (Structured Query Language )

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



**Data Manipulation Language:** SQL'in satır ekle, sil , değiştir komutlarını içerir. (Insert, Update, Delete, Trigger)

**Data Definition Language:** SQL tablolarının ve görünüşlerin oluşturulması, silinmesi, ve değiştirilmesi komutlarını içerir. (Create, Drop, Alter, Truncate)


**Data Control Language:** Veritabanı kullanıcılarına nesnelere erişim iznin verilmesini veya kısıtlanmasını sağlar . Security (Grant, Revoke)

**Transaction Control Language:** Hareket Yönetimi ile alakalı olan kısım (Commit, Rollback, Savepoint)

**Data Query Language:** Veritabanında sorgulamayı gerçekleştirir (Select)

# SQL Yazım Kuralları (Syntax)

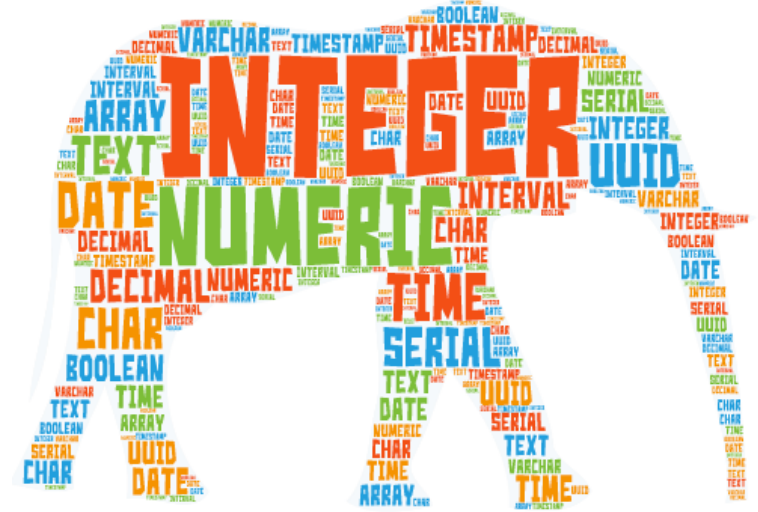
---

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

SQL de SELECT=Select=select

## PostgreSQL Veri Tipleri

- 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)



# PostgreSQL Veri Tipleri: Char, Varchar, Text

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

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

# SQL: Tablo Oluştur: create table

```
CREATE TABLE [IF NOT EXISTS] tabload1 (  
    column1 datatype(length) column_constraint,  
    column2 datatype(length) column_constraint,  
    column3 datatype(length) column_constraint,  
    table_constraints  
);
```

- Not NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

# SQL: Tablo Oluştur: create table

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  
);
```

## SQL: Tablo Sil (drop table)

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;

## SQL: Veri ekleme (insert)

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

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

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



# SQL: Seçim ve Listeleme (select)

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

**Select** \* **from** öğrenci;

Tabloda bulunan  
veriyi listeler

# SQL: Seçim ve Listeleme (select)

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

```
Select kullanıcıadı,ortalama  
from öğrenci;
```



# SQL: Seçim ve Listeleme (select)

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

```
Select kullanıcıadı,ortalama  
from öğrenci;
```

	kullanıcıadı character varying (20)	ortalama numeric
1	Eymen234	3.2
2	Kml45	3.7
3	Cm3418	2.9
4	Hayri95	3.8
5	m.4417	5
6	Mhmt4456	4
7	Arf2332	3.2

# SQL: Şart ile listeleme (select - where)

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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



# SQL: Şart ile listeleme (select - where)

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	16458	Hayri	Hayri95	19	3.8
3	17327	Mustafa	m.4417	18	5
4	17111	Mehmet	Mhmt4456	20	4
5	17123	Arif	Arf2332	24	3.2

# SQL: Sıralama (order)

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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



# SQL: Sıralama (order)

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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

ASC (Artan)

DESC (Azalan)

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17347	Kemal	Kml45	17	3.7
2	17327	Mustafa	m.4417	18	5
3	17236	Cemil	Cm3418	18	2.9
4	16458	Hayri	Hayri95	19	3.8
5	17532	Eymen	Eymen234	19	3.2
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

# SQL: distinct

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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





# SQL: distinct

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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

	yaş integer
1	17
2	18
3	19
4	20
5	24

# SQL: kümeleme fonksiyonları (aggregate functions)

---

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

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

# SQL: avg

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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



# SQL: avg

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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

	avg numeric
1	3.6857142857142857

# SQL: max

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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



# SQL: max

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17327	Mustafa	m.4417	18	5

# SQL: min

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

```
select öğrencino, isim  
from öğrenci  
where ortalama =(  
    select min (ortalama) from öğrenci);
```



# SQL: min

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

```
select öğrencino, isim
from öğrenci
where ortalama = (
    select min (ortalama) from öğrenci);
```

	öğrencino integer	isim character varying (20)
1	17236	Cemil



# SQL: min

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullaniciadi character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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



# SQL: min

Öğrenci Tablosu

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
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
6	17111	Mehmet	Mhmt4456	20	4
7	17123	Arif	Arf2332	24	3.2

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

	ögrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	17347	Kemal	Kml45	17	3.7
3	16458	Hayri	Hayri95	19	3.8
4	17327	Mustafa	m.4417	18	5
5	17111	Mehmet	Mhmt4456	20	4
6	17123	Arif	Arf2332	24	3.2
7	17236	Cemil	Cm3418	18	3.9

# SQL: min

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	17347	Kemal	Kml45	17	3.7
3	16458	Hayri	Hayri95	19	3.8
4	17327	Mustafa	m.4417	18	5
5	17111	Mehmet	Mhmt4456	20	4
6	17123	Arif	Arf2332	24	3.2
7	17236	Cemil	Cm3418	18	3.9

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

```
Select * from öğrenci;
```



# SQL: min

## Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	17347	Kemal	Kml45	17	3.7
3	16458	Hayri	Hayri95	19	3.8
4	17327	Mustafa	m.4417	18	5
5	17111	Mehmet	Mhmt4456	20	4
6	17123	Arif	Arf2332	24	3.2
7	17236	Cemil	Cm3418	18	3.9

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

```
Select * from öğrenci;
```

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	17347	Kemal	Kml45	17	3.7
3	16458	Hayri	Hayri95	19	3.8
4	17327	Mustafa	m.4417	18	5
5	17111	Mehmet	Mhmt4456	20	4
6	17123	Arif	Arf2332	24	3.2
7	17236	Cemil	Cm3418	18	3.9

# SQL: min

Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	17347	Kemal	Kml45	17	3.7
3	16458	Hayri	Hayri95	19	3.8
4	17327	Mustafa	m.4417	18	5
5	17111	Mehmet	Mhmt4456	20	4
6	17123	Arif	Arf2332	24	3.2
7	17236	Cemil	Cm3418	18	3.9

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

```
Select * from öğrenci;
```



# SQL: min

## Öğrenci Tablosu

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	17347	Kemal	Kml45	17	3.7
3	16458	Hayri	Hayri95	19	3.8
4	17327	Mustafa	m.4417	18	5
5	17111	Mehmet	Mhmt4456	20	4
6	17123	Arif	Arf2332	24	3.2
7	17236	Cemil	Cm3418	18	3.9

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

	öğrencino integer	isim character varying (20)	kullanıcıadı character varying (20)	yaş integer	ortalama numeric
1	17532	Eymen	Eymen234	19	3.2
2	16458	Hayri	Hayri95	19	3.8
3	17327	Mustafa	m.4417	18	5
4	17111	Mehmet	Mhmt4456	20	4
5	17123	Arif	Arf2332	24	3.2
6	17236	Cemil	Cm3418	18	3.9

## SQL Statement:

```
SELECT * FROM Customers;
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

## Result:

Number of Records: 91

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	Berguvsvägen 8	Luleå	S-958 22	Sweden

## Your Database:

Tablename	Records
<a href="#">Customers</a>	91
<a href="#">Categories</a>	8
<a href="#">Employees</a>	10
<a href="#">OrderDetails</a>	518
<a href="#">Orders</a>	196
<a href="#">Products</a>	77
<a href="#">Shippers</a>	3
<a href="#">Suppliers</a>	29

Restore Database

## Retrieve everything from a table



### Question

How can you retrieve all the information from the cd.facilities table?

Schema reminder ▲

**cd.members**

memid	integer
surname	character varying(200)
firstname	character varying(200)
address	character varying(300)
zipcode	integer
telephone	character varying(20)
recommendedby	integer
joindate	timestamp

**cd.bookings**

facid	integer
memid	integer
starttime	timestamp
slots	integer

**cd.facilities**

facid	integer
name	character varying(100)
membercost	numeric
guestcost	numeric
initialoutlay	numeric
monthlymaintenance	numeric

### Expected Results

facid	name	membercost	guestcost	initialoutlay	monthlymaintenance
0	Tennis Court 1	5	25	10000	200
1	Tennis Court 2	5	25	8000	200
2	Badminton Court	0	15.5	4000	50
3	Table Tennis	0	5	320	10
4	Massage Room 1	35	80	4000	3000
5	Massage Room 2	35	80	4000	3000

### Your Answer

Hint

Help

Save

Run Query

```
select * from cd.facilities;
```



Dinlediğiniz İçin  
Teşekkürler....

---