

Veritabanı Yönetim Sistemleri

Dr. Öğr. Üyesi Ahmet Arif AYDIN

İlişkisel Hesap (Relational Algebra)
SQL-1

- 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

Logic: Mantık

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

Dönüşüm Kuralları (Transformation Rules)

ifade1 eşittir ifade2 ($\text{ifade1} \equiv \text{ifade2}$)

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

Implication Law

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)
- Örnegin
 - P = Hava kapalıdır
 - Q = Güneş görünmemektedir

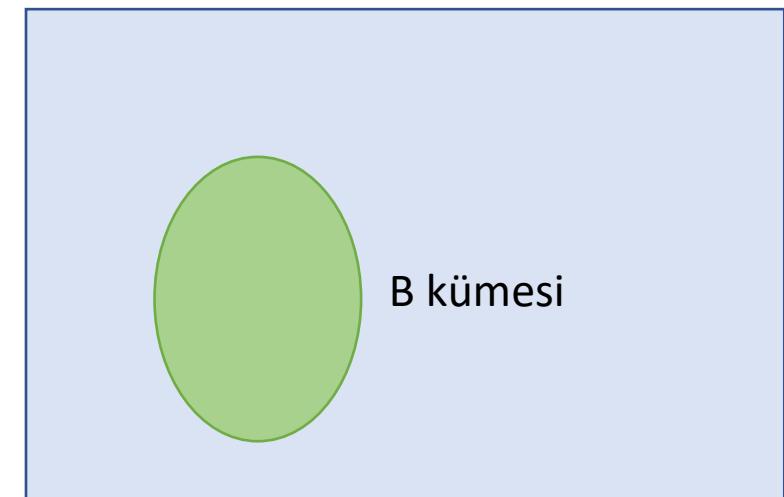
$P \rightarrow Q$ eşit degildir $Q \rightarrow P$

Double Negation Law

Bir ifadenin **değilinin değili (olumsuzun olumsuzu)** kendisidir.

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

Evrensel Küme



De Morgan Kuralı

DeMorgan's Law

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

Distributive Law

Dağıtım Kuralı

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

İlişkisel Hesap (Relational Calculus)

- İlişkisel cebirin (relational algebra) bir alternatifidir.
- Declarative dir.
- Cevapların nasıl hesaplanacağının detayları verilmez (nonprocedural)
- SQL dilinin ortaya çıkmasında büyük etkisi olmuştur.

İlişkisel Hesap (Relational Calculus)

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

Tuple Relational Calculus (TRC)

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

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

Tuple Relational Calculus (TRC)

- { t | şart(t) }
- Şart:
 - Atomik formül
 - $\neg P$ = not P (p nin değil)
 - $P \wedge Q$ = P ve Q
 - $(P \vee Q)$ = P veya Q
 - $P \rightarrow Q$ = P doğru ise Q da doğrudur
 - $\exists R$ (şart (R)) : (\exists : en az bir tane var(**there exists**))
 - $\forall K$ (şart (K)) (\forall : hepsi için (for all))

Domain Relational Calculus (DRC)

- $\{ < \text{sütun1}, \text{sütun2}, \text{sütun3}, \dots \text{sütun n} > \mid P(\text{sütun1}, \text{sütun2}, \text{sütun3}, \dots \text{sütun n}) \}$
- Hesaplanan sonuç: tuple(satır) yerine seçilen sutunlardır.
- $\{ < \text{işçi.no}, \text{işçi.adı} > \mid < \text{işçi.no}, \text{işçi.adı} > ? \text{ İşçi} \wedge \text{bölüm.no} = 10 \}$
 - 10 nolu bölümde çalışan işçilerin adını ve işçi nosunu getir.

Örnek

Kullanıcı Reyting

| 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

Örnek

Reytingi 0 olan kitapları bul

{ t | şart(t) }

{ T | $K \in \text{KullanıcıReyting} \wedge K.\text{reyting} = 0$ }

| 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 |

Örnek

Reytingi 5 den büyük olan kitapların isbn numaralarını
ve reytingi veren kullanıcının alanlarını getir.

{ t | şart(t) }

{ T | $\exists K$ **KullanıcıReyting**

$(K.reyting > 5 \wedge$

$T.kullanıcı = K.kullanıcıno \wedge$

$T.isbn = K.isbn)$ }

| 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 |

Örnek

Reytingi 5 den büyük olan kitapların isbn numaralarını
ve reytingi veren kullanıcının alanlarını getir.

{ t | şart(t) }

{ T | $\exists K$ **KullanıcıReyting**

$(K.reyting > 5 \wedge$

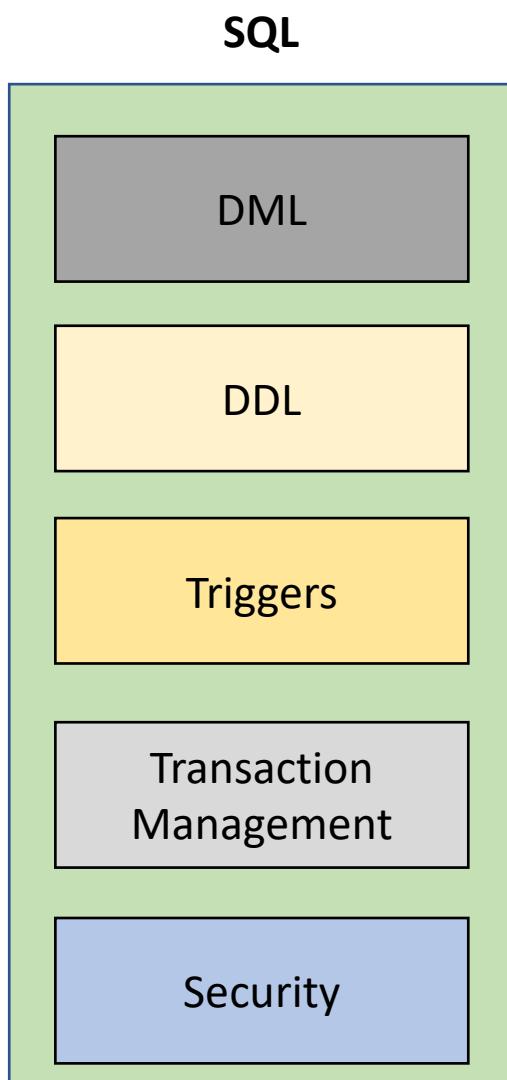
$T.kullanıcı = K.kullanıcı \wedge$

$Tisbn = K.isbn)$ }

| Tkullanıcı | Tisbn |
|------------|------------|
| 276747 | 0060517794 |
| 276751 | 3596218098 |
| 276754 | 0684867621 |

| 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)



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 nesnelerine erişim denetimi sağlanır.

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 |

SQL: Create Table

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 |

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

SQL: Drop Table

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 |

Bir tabloyu silmek için

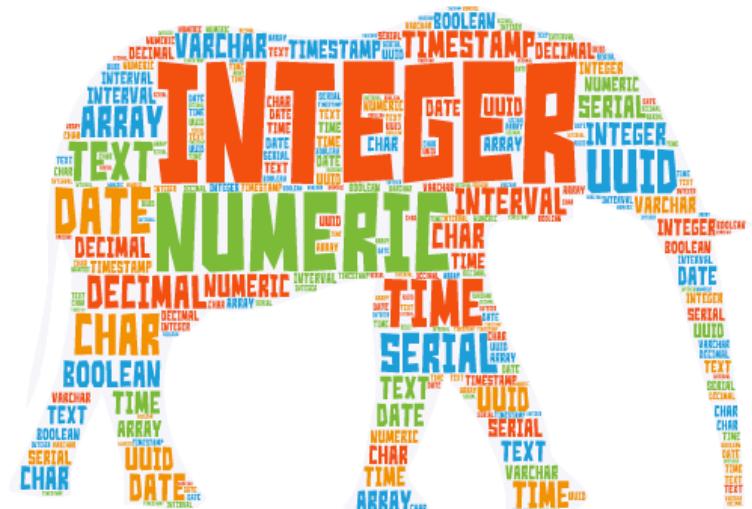
DROP TABLE öğrenci;

SQL Syntax: Yazım Kuralları

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

PostgreSQL Data Types: Veri Tipleri

- Boolean (True-False)
- Karakter Tipi (char,varchar,text)
- Sayısal (integer, floating-point)
- 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 Data Types: Veri Tipleri

Array

```
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' ]
);
```



PostgreSQL Data Types: Veri Tipleri

JSON

```
1 CREATE TABLE orders (
2   ID serial NOT NULL PRIMARY KEY,
3   info json NOT NULL
4 );
```

```
INSERT INTO orders (info)
VALUES
(
  '{ "customer": "John Doe", "items": {"product": "Beer", "qty": 6}}
);
```

PostgreSQL Data Types: Veri Tipleri

- **hstore**: sutun içerisinde bir anahtar-> değer yapısı oluşturmayı sağlar

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

```
1 INSERT INTO books (title, attr)
2 VALUES
3 (
4 'PostgreSQL Tutorial',
5 "paperback" => "243",
6 "publisher" => "postgresqltutorial.com",
7 "language" => "English",
8 "ISBN-13" => "978-1449370000",
9 "weight" => "11.2 ounces"
10 );
```

PostgreSQL Data Types: Veri Tipleri

| 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 |

<https://www.postgresql.org/docs/9.1/static/datatype-character.html>

SQL: Veri Ekleme (Insert Into)

| ögrencino | 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 (ögrencino, isim, kullaniciadi, yas, ortalama) values (17532, 'Eymen', 'Eymen234',19,3.2);
```

SQL: Seçim ve Listeleme (SELECT)

| 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 |

Tabloda bulunan
veriyi listeler

Select * FROM öğrenci;

SQL: Seçim ve Listeleme (SELECT)

| 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 ögrencino, isim from öğrenci;
```

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

SQL: Şart ile listeleme (Select + Where)

| 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  
    yaşı > 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 |

SQL: Sıralama (ORDER)

| 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  
order by yaş 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)

SQL: Distinct

| 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 distinct yaş from öğrenci order by yaş;

| | yas |
|---|---------|
| | integer |
| 1 | 17 |
| 2 | 18 |
| 3 | 19 |

SQL: Aggregate Functions

Birden fazla satırdan sonuç olarak
sadece bir satır üreten fonksiyonlara aggregate functions denir

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

SQL:avg (ortalama)

| 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 avg (ortalama) from öğrenci ;

| | |
|---|-------------------------|
| | avg double precision |
| 1 | 3.52000002861023 |

SQL: max

| 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 |

SQL:min

| ögrencino | isim | kullanıcıadi | 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 | kullanıcıadi character varying | yas integer | ortalama real |
|---|----------------------|---------------------------|-----------------------------------|----------------|------------------|
| 1 | 17236 | Cemil | Cm3418 | 18 | 2.9 |

SQL: Update

| ögrencino | isim | kullanıcıadi | 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 | kullanıcıadi 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 |

SQL: Delete

| öğrencino | isim | kullanıcıadi | 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) ;