



SQL'e GİRİŞ

Çağlar Ülküderner

caglar@tubitak.gov.tr

Database yaratmanın zorlukları?

- Ah o data structure dersi..
- Eskiden bir kaç text yada binary dosya üzerinde herşeyi yapmaya çalışırdık.
- Fonksiyonlar yazıp sevinirdik.
- Buffer ayarlamaya çalışırdık.

Database Neden?

- Çok geniş kütüklerle boğuşmak
 - İhtiyaçlar çok fazla
 - Kütüklerin yönetimi
 - Kütükleri korumak
 - Yedekleme
 - Güvenlik

Database Neden?

- Birden fazla veri yapısını bir arada tutmak
- Ya ekleme yapmak gerekirse
- Peki ya değişiklik olursa
- Bu dosya yapıları çöpe mi yoksa bir beyin fırtınısından ve yüzlerce satırlık kod yazdıktan sonra bir sonraki fırtınaya kadar geçici bir rahatlama mı?

Peki ya internet yüzü?

- Online gezinirken gerekli olursa
 - Bir database in ulaşılabilirliği
 - Sunucu gücü
 - Hazırlanılan bu dosyalara ulaşımı neyle yapacağız?
 - Peki ya güvenlik?

Peki SQL Nedir?

- Structured Query Language
- Database yönetimi için bir programlama standardı.
- ANSI ve ISO standartlarında ve bir çok programlama dilli standart komutlarında SQL özelliklerini kullanmaya imkan sağlıyor.

SQL Kullanımı nasıl?

- Basitçe alışılabilecek komutlarla kullanım
- Veri ekleyip çıkarmaktan tutup
- Veri yerini kolaylıkla bulmaya kadar
- Programlanabilir arayüz mevcut
- Yedek almak ve alınan yedekleri geri yüklemek çok kolay

SQL Nerede Lazım?

- Aslında her eve lazım :)
- Şirketin mali durumlarını toplarlamak
- Stok kontrolleri
- Veri arama sistemleri (Arama moturu, vb)
- Özellikle PHP kullananlara

SQL Ağzıyla kuş tutar mı?

- Aslında bu biraz da kullandığınız SQL'e bağlı (MySQL, PostgreSQL vs.)
- Başlangıç için en iyisi MySQL
- Sonraki adım PostgreSQL
- Tüm ihtiyaçlarınıza ücretsiz çözüm

MySQL ne? Yenir mi?

- İlk başlayanlar için MySQL
 - Altın dörtlünün bir parçası
- MyAdmin ile zorlanmadan öğrenerek kullanım imkanı
- Çoklu dil seçeneği ile kolay kullanım

MySQL ile ısınma hareketleri

■ MySQL'e bağlanmak

```
mysql -h localhost -u root -p
```

```
Enter password: *****
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 18 to server version: 3.23.38
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer
```

```
mysql>
```

```
mysql>
```

MySQL ile ısınma hareketleri

- **MySQL değişkenlerini görmek**

mysql> show variables;

- Versiyon bilgisinden bölge saatine kadar 78 farklı bilgi verir.

mysql> status;

Connection id:	18
Current database:	
Current user:	root@localhost
Server version:	3.23.38
Protocol version:	10
Connection:	localhost via TCP/IP
Client characterset:	latin1
Server characterset:	latin1
TCP port:	3306
Uptime:	11 min 43 sec

Threads: 2 Questions: 247 Slow queries: 0 Opens: 243 Flush tables:
1 Open tables: 64 Queries per second avg: 0.351

MySQL ile ısınma hareketleri

- Databaseleri görmek:

```
mysql> show databases;
```

```
+-----+  
| Database |  
+-----+  
| cengdb   |  
| deneme   |  
| muhasebe |  
| mysql    |  
| nuke     |  
| test     |  
| theatre  |  
+-----+
```

```
7 rows in set (0.00 sec)
```

```
show databases;
```

Sisteme bağlı olan kullanıcının haklarının izin verdiği tüm databaseleri gösterir.

MySQL ile ısınma hareketleri

- Database seçmek için:

`use <database adı>`

Ör:

`mysql> use cengdb;`

Database changed

- Database içindeki tableları görmek için:

`show tables;`

[Veri tabanında veriler tablolar halinde tutulur.]

MySQL ile ısınma hareketleri

```
mysql> show tables;
```

```
+-----+  
| Tables_in_cengdb |  
+-----+  
| course_news      |  
| courses           |  
| exam_news         |  
| exam_types        |  
| firm              |  
| general_news      |  
| instructor         |  
| place_exam_rel    |  
| places            |  
| student           |  
| system_log        |  
| user              |  
+-----+
```

```
12 rows in set (0.00 sec)
```

MySQL ile ısınma hareketleri

■ Tabloyu Analiz etmek:

```
mysql> analyze table courses;
```

Table	Op	Msg_type	Msg_text
cengdb.courses	analyze	status	OK

1 row in set (0.00 sec)

```
mysql>
```


MySQL ile ısınma hareketleri

- Tabloyu optimize etmek:

```
mysql> optimize table courses;
```

```
+-----+-----+-----+-----+
| Table          | Op          | Msg_type | Msg_text |
+-----+-----+-----+-----+
| cengdb.courses | optimize    | status   | OK        |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

MySQL ile ısınma hareketleri

- Tabloyu Onarmak:

```
mysql> repair table courses;
```

Table	Op	Msg_type	Msg_text
cengdb.courses	repair	status	OK

```
1 row in set (0.00 sec)
```

```
mysql>
```

MySQL ile ısınma hareketleri

- Tabloyu kontrol etmek:

```
mysql> check table courses;
```

Table	Op	Msg_type	Msg_text
cengdb.courses	check	status	OK

```
1 row in set (0.00 sec)
```

```
mysql>
```

MySQL ile ısınma hareketleri

- Database yaratma;

`create database <database adı>`

`mysql> create database deneme;`

`Query OK, 1 row affected (0.00 sec)`

Değişken tipleri

TINYINT(M) -> Numerik tamsayılar için. -128, 127 arasında değer alabilir. UNSIGNED ifadesi kullanılabilir.

SMALLINT(M) -> Numerik tamsayılar için. -32768, 32767 arasında değer alabilir. UNSIGNED ifadesi kullanılabilir.

MEDIUMINT(M) -> Numerik tamsayılar için. -8388608, 8388607 arasında değer alabilir. UNSIGNED ifadesi kullanılabilir.

INT(M), INTEGER(M) -> Numerik tamsayılar için. -8388608, 8388607 arasında değer alabilir. UNSIGNED ifadesi kullanılabilir.

BIGINT(M) -> Numerik tamsayılar için. -9223372036854775808, 9223372036854775807 arasında değer alabilir.

FLOAT((M)(D)) -> Üstel tamsayılar için. -3.402823466E+38, -1.175494351E-38 arasında değer alabilir.

Değişken tipleri

DOUBLE((M)(D)) -> Üstel tamsayılar için.
-1.7976931348623157E+308, -2.2250738585072014E-308 arasında
değer alabilir.

DECIMAL((M)(D)) -> Noktalı sayılar için.

DATE -> Tarih formatlı veri için. 'YYYY-MM-DD' formundadır.

DATETIME -> Tarih-saat formatlı veri için. 'YYYY-MM-DD HH:MM:SS'
formundadır.

TIMESTAMP -> Özel tarih-saat formatlı veri için.

TIME -> Zaman formatlı veri için. 'HH:MM:SS' formundadır.

Değişken tipleri

YEAR(2|4) -> 2 yada 4 dijital yıl değeri tutan tarih formatı. 'YYYY' veya 'YY' formundadır.

CHAR(M) -> Sabit uzunluklu karakter dizisi için. 1-255 arasında karakter tutabilir.

VARCHAR(M) -> Değişken uzunluklu karakter dizisi için. 1-255 arasında karakter tutabilir.

Değişken tipleri

TEXT -> Değişken uzunluklu karakter dizisi için. 1-65535 arasında karakter tutabilir.

MEDIUMTEXT -> Değişken uzunluklu karakter dizisi için. 1-16777215 arasında karakter tutabilir.

LONGTEXT -> Değişken uzunluklu karakter dizisi için. 1-4294967295 arasında karakter tutabilir.

ENUM('deger1'..) -> Kullanıcının belirttiği anahtarları tutar. Anahtar 65535'e kadar olabilir.

Table Yaratma

create table <table adı> (degiskenler)

```
CREATE TABLE `tablom` (  
  `id` INT NOT NULL AUTO_INCREMENT,  
  `isim` VARCHAR(25) NOT NULL,  
  `soyisim` VARCHAR(50) NOT NULL,  
  INDEX (`id`),  
  UNIQUE (`id`)  
);
```

Alanların Özelliklerini görme

```
mysql> show columns from tablom;
```

Field	Type	Null	Key	Default	Extra
id	int(11)		PRI	NULL	auto_increment
isim	varchar(25)				
soyisim	varchar(50)				

```
3 rows in set (0.01 sec)
```

Alan özellikleri değiştirme

```
ALTER TABLE `deneme`.`tablom` CHANGE `id` `id`  
BIGINT(11) NOT NULL AUTO_INCREMENT;
```

```
mysql> show columns from tablom;
```

Field	Type	Null	Key	Default	Extra
id	bigint(11)		PRI	NULL	auto_increment
isim	varchar(25)				
soyisim	varchar(50)				

rows in set (0.00 sec)

Tabloya Veri eklemek ve ekli verileri görmek

```
INSERT INTO `tablom`
        (`id`, `isim`, `soyisim`)
VALUES ('1', 'çağlar', 'ülküderner');
```

```
mysql> select * from tablom;
```

```
+-----+-----+-----+
| id | isim      | soyisim    |
+-----+-----+-----+
|  1 | çağlar    | ülküderner |
+-----+-----+-----+
```

```
1 row in set (0.00 sec)
```

Veri eklemek

```
INSERT INTO `tablom`  
          (`id`, `isim`, `soyisim`)
```

```
VALUES ('', 'doruk', 'fişek');
```

```
mysql> select * from tablom;
```

```
+-----+-----+-----+  
| id | isim      | soyisim      |  
+-----+-----+-----+  
|  1 | çağlar    | ülküderner   |  
|  2 | doruk     | fişek        |  
+-----+-----+-----+
```

```
2 rows in set (0.00 sec)
```

Belirli alanları tarama

```
mysql> select * from `tablom` where  
      `isim`='doruk';
```

```
+-----+-----+-----+  
| id | isim | soyisim |  
+-----+-----+-----+  
|  2 | doruk | fişek   |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

Yanlış girilenlerin düzeltilmesi

```
UPDATE `tablom` SET `isim` = 'çağlar 2' WHERE `id` = '1'  
LIMIT 1;
```

```
mysql> select * from tablom;
```

```
+-----+-----+-----+  
| id | isim      | soyisim      |  
+-----+-----+-----+  
| 1 | çağlar2   | ülküderner   |  
| 2 | doruk     | fişek        |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

Distinct ile teke indirgenen seçimler

```
mysql> select * from main;
```

t_id	c_name	t_name	t_row	t_column
1	Ankara	Küçük Tiyatro	12	15
2	Ankara	yeni sahne	10	5
3	İstanbul(Trakya)	bkm	50	20

```
3 rows in set (0.00 sec)
```


Distinct ile indirgenen seçimler

```
mysql> select c_name from  
main;
```

+	-----	+
	c_name	
+	-----	+
	Ankara	
	Ankara	
	İstanbul(Trakya)	
+	-----	+

3 rows in set (0.00 sec)

```
mysql> select distinct  
c_name from main;
```

+	-----	+
	c_name	
+	-----	+
	Ankara	
	İstanbul(Trakya)	
+	-----	+

2 rows in set (0.00 sec)

Distinct ile indirgenen seçimler

```
mysql> select distinct c_name,t_name from  
main;
```

c_name	t_name
Ankara	Küçük Tiyatro
Ankara	yeni sahne
İstanbul(Trakya)	bkm

```
3 rows in set (0.00 sec)
```

Having ile koşula bağlı seçim

```
mysql> SELECT * FROM `tablom`  
      having `id`>1;
```

```
+-----+-----+  
| id | isim  | soyisim |  
+----+-----+-----+  
|  2 | doruk | fişek   |  
+----+-----+-----+  
1 row in set (0.00 sec)
```

Where kullanımı

```
mysql> select * from tablom where  
      id=1;
```

+	-	-	-	+	-	-	-	-	-	-	+	-	-	-	-	-	-	-	+
	id		isim		soyisim														
+	-	-	-	+	-	-	-	-	-	-	+	-	-	-	-	-	-	-	+
	1		çağlar		ülküderner														
+	-	-	-	+	-	-	-	-	-	-	+	-	-	-	-	-	-	-	+

```
1 row in set (0.00 sec)
```

Having + Where kullanımı

```
mysql> SELECT * FROM `tablom` where  
      `id`>0 having `isim`='doruk';
```

```
+-----+-----+-----+  
| id | isim | soyisim |  
+-----+-----+-----+  
|  2 | doruk | fişek   |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

Select biraz daha..

```
mysql> select * from tablom where id between 0  
and 3;
```

+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	+
	id		isim			soyisim										
+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	+
	1		çağlar			ülküderner										
	2		doruk			fişek										
+	-	-	-	-	+	-	-	-	-	-	-	-	-	-	-	+

```
2 rows in set (0.01 sec)
```

Biraz da gerekebilir String fonksiyonları

Like değimi

'_' karakteri => herhangi bir karakter

'%' karakteri => herhangi bir karakter dizisi

```
SELECT 'çağlar' LIKE 'çağ%'
```

1

```
SELECT 'çağlar' LIKE 'çağ__'
```

0

NOT: \ (escape karakteri)

Biraz da gerekebilir String fonksiyonları

' **STRCMP** ' İFADESİ STRCMP (kelime1, kelime2)

- Kelime1 ile Kelime2 aynı ise 0,
- Kelime1 Kelime2'den alafabetik olarak büyük ise 1,
- Diğer durumda -1 degeri üretilir.

SELECT STRCMP('abc','abc') => 0

STRCMP('abc','cba')=> -1

STRCMP('cba','abc') => 1

Biraz da gerekebilir String fonksiyonları

' BINARY ' İFADESİ

SELECT "a"="A"; =>1

SELECT BINARY "a"="A"; =>0

Diğer String fonksiyonları

ASCII (veri) : veri'nin ascii karşılığını bulur.

CONV (veri, a1, a2) : veri'yi a1 tabanlı olarak alıp a2 tabana dönüştürür.

BIN (N) : 10 tabanlı N verisini binary forma yani 2'lik tabana dönüştürür.

OCT (N) : 10 tabanlı N verisini oktal forma yani 8'lik tabana dönüştürür.

HEX (N) : 10 tabanlı N verisini heksadesimal yani 16'lik tabana dönüştürür.

CHAR (N1, N2 ..) : ascii kodlu N1,N2 degerlerini stringe dönüştürür ve birleştirir.

CONCAT (N1, N2 ..) : N1,N2 verilerini stringe çevirip birleştirerek geri verir. NULL değer girilirse sonuç NULL olur.

LENGTH (N1) : N1 stinginin karakter sayısını bulur.

Matematiksel ifadeler

ABS (x) : x'in pozitif değerini bulur.

SIGN (x) : x pozitif ise 1, negatif ise -1, 0 ise 0 değerini üretir.

MOD (x, y) : x y'ye bölünür ve kalan değer fonksiyondan döner.

FLOOR (x) : x'e en yakın ve x'den küçük tamsayıyı bulur.

CEILING (x) : x'e en yakın ve x'den büyük tamsayıyı bulur.

ROUND (x) : x'e en yakın tamsayıyı verir.

ROUND (x, D) : noktalı sayılarda x'e en yakın, noktadan sonra D rakam olan sayıyı verir.

Matematiksel Fonksiyonlar

POW (x, y) , POWER (x, y) : x'in y'ninci kuvvetini bulur.

SQRT (x) : x'in karekökünü bulur.

PI () : Pi sayısını verir.

GREATEST (x, y, z ..) : en büyük veriyi bulur.

TRUNCATE (x, D) : noktalı x sayısını noktadan sonra D rakamlı hale sokar.

Tarih – Saat fonksiyonları

DAYOFWEEK(tarih) => YYYY - MM - DD formudaki tarih'in, ODBC standardında haftanın kaçınıcı günü olduğunu bulur.

SELECT DAYOFWEEK('2001-12-26') =>4

DAYOFMONTH(tarih) => Ayın kaçınıcı günü olduğunu bulur.

DAYOFYEAR(tarih) => Yılın kaçınıcı günü olduğunu bulur.

MONTH(tarih) => Kaçınıcı gün olduğunu bulur.

DAYNAME(tarih) => Gün adını bulur.

MONTHNAME(tarih) => Ay adını bulur.

QUARTER(tarih) => Tarihin, yılın kaçınıcı çeyreğinde olduğunu bulur.

WEEK(tarih) => Yılın kaçınıcı haftası olduğunu bulur.

YEAR(tarih) => Hangi yıl olduğunu bulur.

hour(zaman) => HH : MM : SS formatındaki zaman verisinden saat bulunur.

MINUTE(zaman) => Dakikayı bulur.

SECOND(zaman) => Saniyeyi bulur.

DATE_FORMAT

%M : Ay adi (January, December)

%W : Gun adi (Sunday, Saturday)

%D : Gun degeri ve eki (1st, 2nd)

%Y : Yil degeri , 4 dijit (1993)

%y : Yil degeri , 2 dijit (93)

%a : Kisaltilmis gun adi (Sun, Sat)

%d : Gun degeri , 2 dijit (00 , 31)

%e : Gun degeri (0 , 31)

%m : Ay degeri , 2 dijit (01 , 12)

%c : Ay degeri (1 , 12)

%b : Kisaltilmis ay adi (Jan , Dec)

%j : Yilin gun sayisi (001 , 366)

%H : Saat , 2 dijit (00 , 23)

%k : Saat (0 , 23)

%h : Saat , 2 dijit (01 , 12)

%l : Saat (1 , 12)

%i : Dakika (00 , 59)

%r : Zaman , 12 saatlik (hh : mm : ss
[AP] M)

%T : Zaman , 24 saatlik (hh : mm : ss)

%S : Saniye (00 , 59)

%p : AM yada PM

%w : Haftanin kacinci gunu oldugu
(0=Sunday , 6=Saturday)

%U : Yilin kacinci haftasi oldugu
(Sunday haftanin birinci gunu)

%u : Yilin kacinci haftasi oldugu
(Monday haftanin birinci gunu)

%% : Sadece ' % ' karakteri



DATE SON

'CURDATE ' , ' CURRENT_DATE ' O anki tarih

' NOW ' , ' SYSDATE ' , ' CURRENT_TIMESTAMP ' O anki zaman gün
ay yıl

SELECT NOW() ;

2001-12-26 14:52:46