

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİSİ

AÇIK KAYNAK KODLA VERİ TABANI İŞLEMLERİ

Ankara, 2013

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. VERİ TABANINA GİRİŞ	3
1.1. MS-DOS Komut Satırından MySQL Veri Tabanına Erişme.....	3
1.2. Veri Tabanının Adını Öğrenme	5
1.3. Veri Tabanı Oluşturmak.....	5
1.4. Veri Tabanını Silme	6
1.5. Veri Tabanını Seçme.....	7
1.6. Veri Tabanına Tablo Ekleme	8
1.7. Tablodan Bilgileri Sorgulama Komutu	9
1.8. Tablodan Kayıt Silme	11
1.9. Tablo Özelliklerini Görme	14
1.10. Veri Tipleri.....	15
1.11. Tabloya Alan Ekleme.....	16
1.12. Tablodan Alan Silme	17
1.13. Tablo Adını Değiştirme	18
1.14. Sorgulama Operatörleri.....	18
1.15. Koşula Bağlı Sorgulama Yapma	19
1.16. Verileri Güncelleme.....	22
1.17. Hazır Fonksiyonlar.....	23
UYGULAMA FAALİYETİ	26
ÖLÇME VE DEĞERLENDİRME	27
ÖĞRENME FAALİYETİ-2	28
2. PHP İLE MYSQL PROGRAMLAMA	28
2.1. Veri Tabanı Programlarıyla Çalışma	28
2.2. Yeni Bir Veri Tabanı Oluşturma.....	29
2.3. Veri Tabanı Sunucusuna Bağlantı Kurma.....	33
2.4. Tabloya Bilgi Ekleme	35
2.5. Tablodan Bilgileri Okutma	37
2.6. Tablodan Bilgileri Düzenleme	38
UYGULAMA FAALİYETİ	41
ÖLÇME VE DEĞERLENDİRME	42
MODÜL DEĞERLENDİRME	43
CEVAP ANAHTARLARI.....	45
KAYNAKÇA	46

AÇIKLAMALAR

ALAN	Bilişim Teknolojileri
DAL/MESLEK	Web Programcılığı
MODÜLÜN ADI	Açık Kaynak Kodla Veri Tabanı İşlemleri
MODÜLÜN TANIMI	Bu modül açık kaynak kodlu internet programlama dilini kullanarak veri tabanı işlemlerini yapabileceğiniz öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Açık Kaynak Kodla Grafik İşlemleri ve Nesne Yönelimli Programlama modülünü tamamlamış olmak
YETERLİK	Veri tabanı işlemlerini yapmak
MODÜLÜN AMACI	Genel Amaç Bu modül ile gerekli ortam sağlandığında veri tabanı işlemlerini yapabileceksiniz. Amaçlar <ol style="list-style-type: none">1. Veri tabanı temel düzenlemelerini yapabileceksiniz.2. Veri tabanı programlamasını yapabileceksiniz.
EĞİTİM ORTAMLARI VE DONANIMLARI	Ortam: Bilgisayar laboratuvarı Donanım: Açık kaynak kod tabanlı web yazılımları, bilgisayar
ÖLÇME VE DEĞERLENDİRME	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Gelişen teknoloji ile veri ve bilgilere erişim günümüzde çok kolaylaştı. Önemli olan çok veri kullanmak değil onları doğru zamanda ve doğru yerde kullanmak önemlidir.

Günlük hayatta veri kavramı o kadar çok çoğaldı ki artık bizlere ait birçok bilgi, kayıt altına alındı bile. En basitinden e-okul sisteminde bize ait olan bilgiler; nüfus bilgileri, hastane kayıt bilgileri, LYS puanı vs. bunlar bizim bir yerlerde tutulan bilgilerimizdir.

Peki, sıkça kullandığımız bu veriler nasıl kaydolur, nasıl bulunur, nasıl düzenlenir?

Bu modülümüzde açık kaynak kodlu internet programa dili ile veri tabanında işlemler yapmayı öğreneceksiniz.

ÖĞRENME FAALİYETİ-1

AMAÇ

Bu faaliyet sonunda veri tabanında temel düzenlemeleri yapabileceksiniz.

ARAŞTIRMA

- Veri tabanı sunucuları nelerdir? Araştırınız.
- Veri tabanı sunucularının kullanım mantığı nedir?
- Piyasada kullanılan veri tabanı sunucu programları hangileridir?

1. VERİ TABANINA GİRİŞ

Bilgisayarın en önemli görevlerinden sayılan verileri saklama, verileri işleme ve gerektiğinde ise bu verileri çağırarak kullanma, gündelik hayatımızda birçok işimizi kolaylaştırmaktadır.

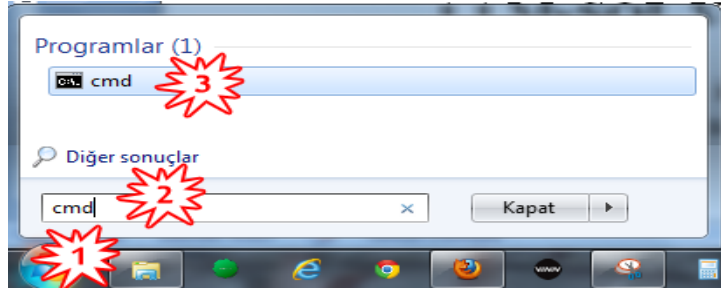
E-okulda notlarımızın ve devamsızlıklarımızın takibi, ÖSYM sınav ve puan bilgilerimiz, sizlerin kullandığı verisel işlemlerdir. İşte, bu tür veriler ve dahası bilgisayarda bir sistem dâhilinde tutulur, çağrılır ve işlenir. Bilgisayarda verilerle uğraşmak için programlar yapılmıştır. Bunlardan MySQL, Oracle, MS-SQL, Access bunlardan en bilinenidir.

Bu öğrenme faaliyetinde, dersimizin ilk modülü olan “Açık Kaynak Kodla Temel Uygulamalar” modülünde kurulumu anlatılmış olan My-SQL veri tabanına erişim ve veri tabanında temel işlemleri gerçekleştireceğiz.

1.1. MS-DOS Komut Satırından MySQL Veri Tabanına Erişme

MS-DOS komut sisteminden, veri tabanına erişebilmek için veri tabanımızın olduğundan emin olmamız gerekmektedir.

Windows işletim sistemi üzerinden Başlat→Çalıştır→cmd komutu ile MS-DOS komut ekranını çalıştırıyoruz.



Resim 1.1. MS-DOS komut satırının başlatılması

Komut satırı aşağıdaki ekran görüntüsündeyken

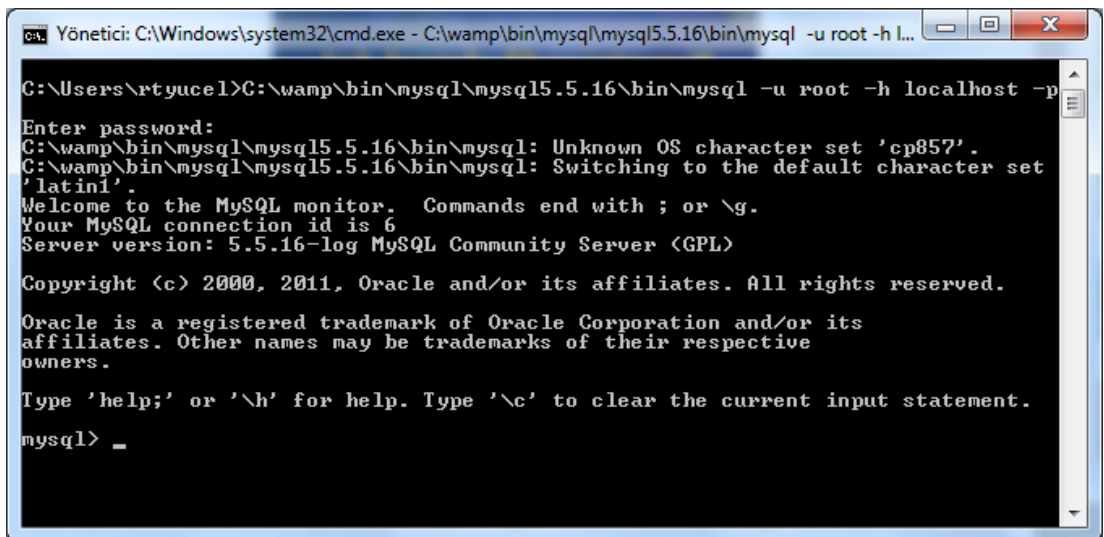


Komut satırına;

```
C:\wamp\bin\mysql\mysql5.5.16\bin\mysql -u root -h localhost -p
```

komutunu yazıp ENTER tuşuna basıyoruz. “Enter Password:” kısmına eğer veri tabanına erişmek için daha önceden belirlenmiş bir şifremiz varsa bu şifreyi giriyoruz yoksa sadece ENTER tuşuna basıyoruz.

Aşağıdaki ekran görüntüsündeki gibi bir ekran karşımıza geldiğinde artık MS-DOS ortamında veri tabanına erişmiş bulunuyoruz ve artık komutlarla işlem yapabiliriz.



1.2. Veri Tabanının Adını Öğrenme

SHOW DATABASES; komutu bize var olan veri tabanlarının listesini sunar. Örnek uygulamada olduğu gibi komut satırında buna bir bakalım.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)
```

Burada yaptığımız sorguda şu an üç tane kurulumla gelen veri tabanı olduğunu görüyoruz.

1.3. Veri Tabanı Oluşturmak

Yeni bir veri tabanı oluşturmak istenirse **CREATE DATABASE VERITABANI_ADI;** komutunun kullanmalıyız.

Şimdi *mtegm* adında bir veri tabanı oluşturalım. Komutlarımızı yazıyor, en sonuna dikkat ettiğiniz gibi noktalı virgül(;) işaretini eklemeyi unutmuyoruz.

```
mysql> create database mtegm;
Query OK, 1 row affected (0.00 sec)

mysql> _
```

Dikkat ettiyseniz komutun altındaki satırda komutun çalıştırılmasından sonra durum raporunu bizlere verdi.

Query OK ibaresi komutun çalışmasının başarılı bir şekilde tamamlandığını ifade etmektedir.

Şimdi oluşan veri tabanımıza bir bakalım. Hatırlayacağınız gibi mevcut veri tabanları görme komutumuz **SHOW DATABASES;** idi.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mtegm |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)
```

Var olan veri tabanı adında başka bir veri tabanı oluşturmak istersek oluşmayacağını ve hata mesajı alacağımızı göreceksiniz.

```
mysql> create database mtegm;
ERROR 1007 (HY000): Can't create database 'mtegm'; database exists
mysql>
```

Görüldüğü üzere **ERROR** uyarısı verdi. Yukarıdaki *database exists* ibaresi veri tabanının var olduğu anlamına gelmektedir.

1.4. Veri Tabanını Silme

Var olan veri tabanını silmek için *DROP* komutu kullanılır. Kullanırken dikkat edilmesi gereken bu komut, bütün bilgileri ve tabloları geri getirilemez bir şekilde siler.

Şimdi oluşturduğumuz *mtegm* veri tabanını silelim. Komutun kullanımı şu şekildedir:
DROP DATABASE VERITABANI_ADI;

Komutu çalıştırmadan önce kayıtlı veri tabanlarının listesini görelim.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mtegm |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)
```

4 adet veri tabanının olduğunu gördükten sonra *DROP* komutumuzu çalıştırıyoruz.

```
mysql> drop database mtegm;  
Query OK, 0 rows affected (0.01 sec)  
  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
3 rows in set (0.00 sec)
```

Yukarıdaki ekran görüntüsünde de görüldüğü gibi var olan veri tabanımızı sildik. Silme işleminden sonra da silinip silinmediğinin kontrolünü yapmış olduk. Dört olan veri tabanı adedimizin üçe düştüğünü gördük.

1.5. Veri Tabanını Seçme

Tablo konusuna geçmeden önce tablo oluşturmanın mantığından biraz bahsedelim. Veri tabanı işlemlerinde mantık şu şekildedir: Önce veri tabanı oluşturulur sonra da veri tabanı için tablo oluşturmak suretiyle aynı gruptaki bilgiler bir araya toplanır.

Şöyle ki bir alışveriş sitesi için veri tabanı oluşturduz. Veri tabanı ismi AVM olsun. Bu veri tabanında *musteriler* adında bir tablo ve *urunler* adında başka bir tablo oluşturursak veriler belli gruplar dâhilinde bir arada tutulmuş olur.

Musteriler tablosunun altında *isim*, *soyad*, *adres*, *telefon*, *adres* gibi müşteriye ait bilgiler tutulurken, *Urunler* adlı tabloda ise *cinsi*, *fiyatı*, *adedi*, *indirim oranı* gibi ürüne ait bilgiler tutulur. Bu iki tablo ise AVM veri tabanının içinde yer alır.

Şimdi bu anlattıklarımızı uygulamalı olarak yapmaya çalışalım: Önce AVM adında bir veri tabanı oluşturuyoruz. Sonra mevcut veri tabanlarından AVM olanı seçiyoruz. Demek ki tablo oluşturacağımız veri tabanını önce seçmemiz gerekiyor.

Bunu da **USE VERITABANI_ADI** komutuyla gerçekleştiriyoruz.

```
mysql> create database AVM;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| avm |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

mysql> use AVM;
Database changed
mysql>
```

Şu an üzerinde işlem yapılan veri tabanı AVM. Şimdi veri tabanı içinde tablo var mı yok mu bunun kontrolünü yapalım. Bunun için de **SHOW TABLE** komutunu kullanıyoruz.

```
mysql> show tables;
Empty set (0.07 sec)
```

Görüldüğü gibi AVM veri tabanımızın içinde hiçbir tablo yok. Şimdi sıra veri tabanı içine tablo oluşturma bölümüne geldi.

Musteri				
NO	ISIM(20)	YAŞ	E-MAIL(30)	ŞEHİR(20)

1.6. Veri Tabanına Tablo Ekleme

Yukarıdaki şekilde bir tablo tanımlaması yapalım.

- **Create table musteri**>>Müşteri adında bir tablo tanımlıyor.
- **No integer not null**>>Müşteri numarasının tam sayı (integer) bir değer alacağını ve not null ile boş geçilemeyeceğini içine muhakkak bir değer girilmesi gerektiğini,
- **İsim varchar(20)**>>İsim karakteri için 20 karakterlik bir alanın tahsis edildiğini,
- **Email varchar(30)**>>E-mail için 30 karakterlik bir alanın tahsis edildiğini,
- **Sehir varchar(20)**>>Şehir için 20 karakterlik bir alanın tahsis edildiğini belirtmek için kullanılır.

```
mysql> create table musteriler(no integer not null, isim varchar(20), yas integer, email varchar(30), sehir varchar(20));
Query OK, 0 rows affected (0.01 sec)
```

Tablo oluşturma ve içine veri kaydetme işlemi başarılı bir şekilde tamamlandığına göre tablomuzun oluşup oluşmadığını kontrol edelim. Hatırlanacağı gibi tablo kontrol komutu **SHOW TABLES;** idi. Örneğimizde bunu uygulayalım:

```
mysql> show tables;
+-----+
| Tables_in_avm |
+-----+
| musteriler    |
+-----+
1 row in set (0.00 sec)
```

Oluşturduğumuz bu tablonun içine veri değerleri girmeye sıra geldi. Veri değerlerini aşağıdaki gibi girelim:

MUSTERILER				
NO	ISIM(20)	YAŞ	E-MAIL(30)	ŞEHİR(20)
001	Sinan	33	sinan@meh.gov.tr	Çankırı

```
mysql> insert into musteriler(no,isim,yas,email,sehir) values(001,'Sinan Balcı',33, 'sinan@meh.gov.tr','Çankırı');
Query OK, 1 row affected (0.05 sec)
```

Query OK mesajı işlemimizin başarılı olduğunu gösteriyor. Kaydımızı tabloda görmek için sorgulama işlemi yapalım.

1.7. Tablodan Bilgileri Sorgulama Komutu

```
SELECT alınacak_alanlar FROM tablo_ismi;
```

Örnek: Burada *musteriler* tablomuzdaki bütün verileri listeledik. Burada kullandığımız yıldız (*) tüm alanların alınacağı anlamına gelmektedir.

```
mysql> select * from musteriler;
+----+-----+-----+-----+-----+
| no | isim      | yas | email                | sehir    |
+----+-----+-----+-----+-----+
| 1  | Sinan Balcı | 33  | sinan@meh.gov.tr    | Çankırı  |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Şimdi başka veri girişleri de yaparak tablomuzun içeriğini zenginleştirelim.

```

mysql> insert into musteriler(no,isim,yas,email,sehir) values(002,'Erdal Aktaş',40,
'erdal@meh.gov.tr','Ankara');
Query OK, 1 row affected (0.13 sec)

mysql> select * from musteriler;
+----+-----+-----+-----+-----+
| no | isim          | yas | email                | sehir  |
+----+-----+-----+-----+-----+
| 1  | Sinan Balcı  | 33  | sinan@meh.gov.tr    | Çankırı |
| 2  | Erdal Aktaş  | 40  | erdal@meh.gov.tr    | Ankara  |
+----+-----+-----+-----+-----+
2 rows in set (0.05 sec)

mysql> insert into musteriler(no,isim,yas,email,sehir) values(003,'Deniz Kılıç',50,
'deniz@meh.gov.tr','Adana');
Query OK, 1 row affected (0.03 sec)

mysql> select * from musteriler;
+----+-----+-----+-----+-----+
| no | isim          | yas | email                | sehir  |
+----+-----+-----+-----+-----+
| 1  | Sinan Balcı  | 33  | sinan@meh.gov.tr    | Çankırı |
| 2  | Erdal Aktaş  | 40  | erdal@meh.gov.tr    | Ankara  |
| 3  | Deniz Kılıç  | 50  | deniz@meh.gov.tr    | Adana   |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> _

```

Yukarıdaki örnekte 2 müşteriye ait verileri de tablomuza girmiş olduk.

MUSTERILER				
NO	ISIM(20)	YAS	E-MAIL(30)	ŞEHİR(20)
1	Sinan Balcı	33	sinan@meh.gov.tr	Çankırı
2	Erdal Aktaş	40	erdal@meh.gov.tr	Ankara
3	Deniz Kılıç	50	deniz@meh.gov.tr	Adana

Bu örneğimizde aynı isim, sıra numarası ve bilgileri kaydederek üst üste birden fazla kayıt yapabiliriz.

```

mysql> insert into musteriler(no,isim,yas,email,sehir) values(003,'Deniz Kılıç',50,
'deniz@meh.gov.tr','Adana');
Query OK, 1 row affected (0.00 sec)

mysql> select * from musteriler;
+----+-----+-----+-----+-----+
| no | isim          | yas | email                | sehir  |
+----+-----+-----+-----+-----+
| 1  | Sinan Balcı  | 33  | sinan@meh.gov.tr    | Çankırı |
| 2  | Erdal Aktaş  | 40  | erdal@meh.gov.tr    | Ankara  |
| 3  | Deniz Kılıç  | 50  | deniz@meh.gov.tr    | Adana   |
| 3  | Deniz Kılıç  | 50  | deniz@meh.gov.tr    | Adana   |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

Bir veri tabanında isimler ve bilgiler aynı olabilir ama sıra numaralarının farklı olması gerekmektedir. Bu örnekte 3 sıra numarası iki defa kullanılmıştır. Bu istenilen bir kayıt biçimi değildir. Bunun sebebi tablo tanımlaması yaparken No kısmını *auto increment primary key* (otomatik artan farklı değer) olarak tanımlamak gerekmektedir.

1.8. Tablodan Kayıt Silme

Tablo içindeki kayıtları silmek için **DELETE FROM tablo_adi** komutu kullanılmalıdır. Bazı durumlarda bütün kayıtları değil belirlediğimiz kayıtları silmemiz söz konusudur. Bunun için ise **WHERE** bağlacını ve koşulu ekleriz.

Örnek: Yukarıdaki *musteri* tablosundan şehri *Adana* olan kayıtları silelim.

```
DELETE FROM musteriler WHERE sehir='Adana';
```

```
mysql> delete from musteriler where sehir='Adana';
Query OK, 2 rows affected (0.07 sec)

mysql> select * from musteriler;
+----+-----+-----+-----+-----+
| no | isim      | yas  | email          | sehir   |
+----+-----+-----+-----+-----+
| 1  | Sinan Balcı | 33  | sinan@meh.gov.tr | Çankırı |
| 2  | Erdal Aktaş | 40  | erdal@meh.gov.tr | Ankara  |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> _
```

Bu örnekte AVM veri tabanı içinde bulunan *musteriler* tablosu içindeki memleketi “Adana” olan müşterilerimizi sildik.

Tablodan veri silme örneği daha değişik koşullarla da yapabiliriz.

SİLME ŞARTI	KOMUT
Yaşı 35 den küçük olan müşterileri silmek için	DELETE FROM musteriler WHERE yas<35;
Yaşı 40'a eşit olan müşterileri silmek için	DELETE FROM musteriler WHERE yas=40;
İsmi “Sinan BALCI” olan müşterileri silmek için	DELETE FROM musteriler WHERE isim='Sinan Balcı';
Sıra numarası 2 olan müşteriyi silmek için	DELETE FROM musteriler WHERE no=2;

Tablo içindeki tüm verileri silmek için ise **TRUNCATE TABLE tablo_adi;** komutunu kullanabiliriz.

```
mysql> select * from musteri;
+-----+-----+-----+-----+-----+
| no | isim          | yas | email          | sehir |
+-----+-----+-----+-----+-----+
| 1 | Sinan Balcı | 33 | sinan@meh.gov.tr | Çankırı |
| 2 | Erdal Aktaş | 40 | erdal@meh.gov.tr | Ankara |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> truncate table musteri;
Query OK, 0 rows affected (0.08 sec)

mysql> select * from musteri;
Empty set (0.00 sec)
```

Silme sonucunda tablomuzu sorguladığımızda (*select * from musteri;*) bize tablonun boş olduğu (*Empty*) karşılığı verecektir.

Konumuzun pekişmesi ve diğer konularda da kullanmak için başka bir örnek uygulama yapalım.

Örnek: Okul veri tabanı altına *ogrenciler* diye bir tablo oluşturup öğrenci sınıf, isim, yaş ve cinsiyetlerini kaydedelim.

Şablon tablomuz aşağıdaki şekilde olacaktır:

ogrenci				
Sıra	Sınıf	İsim	Yaş	Cinsiyet
.....

```
mysql> create database okul;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| avm |
| mysql |
| okul |
| performance_schema |
+-----+
5 rows in set (0.07 sec)
```

Veri tabanımız oluştu, şimdi tablo oluşturmaya sıra geldi.

NOT: Tablo oluşturmadan önce hangi veri tabanı kullanılacağı seçilmelidir. Aksi takdirde bizim istediğimiz değil o an atanmış olan veri tabanının içine tablo oluşacaktır.


```
CREATE TABLE ogrenci
(
  no int unsigned not null auto_increment primary key,
  sinif varchar(5),
  isim varchar(30),
  yas int,
  cinsiyet varchar(5)
);
```

```
mysql> use okul;
Database changed
mysql> create table ogrenci(no int unsigned not null auto_increment primary key,
sinif varchar(5), isim varchar(30), yas int, cinsiyet varchar(5));
Query OK, 0 rows affected (0.04 sec)
mysql> _
```

Şimdi oluşturduğumuz bu tabloya aşağıdaki şekilde veri girişleri gerçekleştirelim:

no	sinif	isim	yaş	Cinsiyet
1	9A	Ahmet OKUR	16	ERKEK
2	9A	Zeynep ASLAN	16	KIZ
3	9A	Osman KOÇAK	17	ERKEK
4	10A	Enes AKTAŞ	17	ERKEK
5	10A	Fatma EKTİ	18	KIZ

```
mysql> insert into ogrenci(no,sinif,isim,yas,cinsiyet) values(1,'9A','Ahmet OKUR',16,'ERKEK');
Query OK, 1 row affected (0.04 sec)

mysql> insert into ogrenci(no,sinif,isim,yas,cinsiyet) values(2,'9A','Zeynep ASLAN',16,'KIZ');
Query OK, 1 row affected (0.00 sec)

mysql> insert into ogrenci(no,sinif,isim,yas,cinsiyet) values(3,'9A','Osman KOÇAK',17,'ERKEK');
Query OK, 1 row affected (0.00 sec)

mysql> insert into ogrenci(no,sinif,isim,yas,cinsiyet) values(4,'10A','Enes AKTAŞ',17,'ERKEK');
Query OK, 1 row affected (0.00 sec)

mysql> insert into ogrenci(no,sinif,isim,yas,cinsiyet) values(5,'10A','Fatma EKTİ',18,'KIZ');
Query OK, 1 row affected (0.00 sec)

mysql>
```

Verileri sorguladığımız zaman aşağıdaki şekilde bir görüntü elde etmiş oluruz.

```
mysql> select * from ogrenci;
+----+-----+-----+-----+-----+
| no | sınıf | isim      | yas | cinsiyet |
+----+-----+-----+-----+-----+
| 1  | 9A    | Ahmet OKUR | 16  | ERKEK    |
| 2  | 9A    | Zeynep ASLAN | 16  | KIZ      |
| 3  | 9A    | Osman KOÇAK | 17  | ERKEK    |
| 4  | 10A   | Enes AKTAŞ | 17  | ERKEK    |
| 5  | 10A   | Fatma EKTİ | 18  | KIZ      |
+----+-----+-----+-----+-----+
5 rows in set (0.04 sec)
```

Şimdi bu örnek üzerinde şöyle bir uygulama daha yapalım:

Şimdi de 6. öğrenciyi ekleyelim yalnız numarası “Fatma EKTİ” ile aynı olsun yani numarası 5 olacak şekilde veri girişi yapalım:

```
mysql> insert into ogrenci(no,sinif,isim,yas,cinsiyet) values(5,'10A','Ayşe SEVEN',18,'KIZ');
ERROR 1062 (23000): Duplicate entry '5' for key 'PRIMARY'
mysql> _
```

Burada numarası 5 olan kaydı kabul etmedi çünkü 5 numara başka bir öğrenciye aitti. Daha önceki örnekte ise kaydı kabul etmişti. Bunun sebebi, *no* sütununu *PRIMARY KEY* olarak atamamızdan kaynaklanmaktadır.

1.9. Tablo Özelliklerini Görme

Tanımladığımız tablonun özelliklerini görmek için **DESCRIBE** komutu kullanılır. Bu komutun kullanım şekli aşağıdaki gibidir:

```
DESCRIBE tablo_ismi;
```

Oluşturduğumuz tablonun özelliklerine baktığımızda *NO* alanının *auto_increment* değer aldığını ve aynı değeri bir daha almayan anahtar veriler alacağını öğrenmiş olduk.

```
mysql> DESCRIBE ogrenci;
+----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+----+-----+-----+-----+-----+-----+
| no    | int(10) unsigned | NO   | PRI | NULL    | auto_increment |
| sınıf | varchar(5)      | YES  |     | NULL    |                |
| isim  | varchar(30)     | YES  |     | NULL    |                |
| yas   | int(11)         | YES  |     | NULL    |                |
| cinsiyet | varchar(5)     | YES  |     | NULL    |                |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Aynı zamanda *type* kısmında veri türlerinin neler olduğunu ve kaç karakterlik tahsis yapıldığı gözükmektedir.

1.10. Veri Tipleri

Veri tipleri ve kullanım şekillerinden aşağıdaki bilgileri bilmekte fayda var. Bunlar en çok kullanılan veri tipleridir.

unsigned	Sayı pozitif bir tam sayı olmak zorundadır.
not null	Veri tanımsız (null) olamaz.
auto_increment	Yeni veri eklendiğinde artar.
primary key	Tablonun ana sütunudur ve her zaman farklıdır.
int	Veri sayısal bir değerdir.
varchar(n)	Veri n kadar harf içeren bir alfasayısal değerdir.

Bir veri tabanı içine birden fazla tablo eklemek mümkündür. Bu uygulamamızda “OKUL “ veri tabanı altına eklediğimiz “ogrenci” tablosunun yanına “velibilgi” diye başka bir tablo oluşturalım.

Veli Bilgi Tablosunu oluştururken aşağıdaki tablodan faydalanabiliriz:

No	İsim	İş	Geliri	Adres
ID	Varchar(25)	Varchar(20)	int	Varchar(60)

```
Create table velibilgi(id int unsigned not null auto_increment primary key, isim varchar(25), is varchar(20), gelir int, adres varchar(60), tel int;
```

```
mysql> show tables;
+-----+
| Tables_in_okul |
+-----+
| ogrenci        |
+-----+
1 row in set (0.00 sec)

mysql> create table velibilgi(id int unsigned not null auto_increment primary key, isim varchar(25), isi varchar(20), gelir int, adres varchar(60));
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_okul |
+-----+
| ogrenci        |
| velibilgi      |
+-----+
2 rows in set (0.00 sec)
```

SHOW TABLES; komutu ile oluşan tabloları görebiliyoruz.

Daha önceki örneğimizde “ogrenci” tablosunun içine bilgiler eklemiştik. Şimdi ise “velibilgi” tablosu içine bilgi ekleme işlemi gerçekleştireceğiz.

ID	isim	isi	Geliri	Adres
1	Hasan OKUR	Öğretmen	1900	Varlık Mah.
2	Hasan ASLAN	İşçi	1500	Emek Mah.
3	Recep KOÇAK	Öğretmen	1850	Ekici Mah.
4	Sadık AKTAŞ	Memur	1750	Toki Konutları
5	Sıddık EKTİ	İşçi	1000	Emek Mah.

Yukarıdaki tabloyu kullanarak bilgi girişlerini gerçekleştirelim:

```
mysql> insert into velibilgi(id,isim,isi,gelir,adres) values(1,'Hasan OKUR','öğr
etmen','1900','Varlık Mah.');
```

```
Query OK, 1 row affected (0.07 sec)
```

```
mysql> insert into velibilgi(id,isim,isi,gelir,adres) values(2,'Hasan ASLAN','iş
çi','1500','Emek Mah.');
```

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

```
mysql> insert into velibilgi(id,isim,isi,gelir,adres) values(3,'Recep KOÇAK','öğ
retmen','1850','Ekici Mah.');
```

```
Query OK, 1 row affected (0.06 sec)
```

```
mysql> insert into velibilgi(id,isim,isi,gelir,adres) values(4,'Sadık AKTAŞ','Me
mur','1750','Toki Konutları');
```

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

```
mysql> insert into velibilgi(id,isim,isi,gelir,adres) values(5,'Sıddık EKTİ','iş
çi','1000','Emek Mah.');
```

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

Şimdi de verileri doğru girip girmediğimizin kontrolünü yapalım. Bunun için “**Select * from table;**” komutunu kullanacağız.

```
mysql> select * from velibilgi;
```

id	isim	isi	gelir	adres
1	Hasan OKUR	öğretmen	1900	Varlık Mah.
2	Hasan ASLAN	İşçi	1500	Emek Mah.
3	Recep KOÇAK	öğretmen	1850	Ekici Mah.
4	Sadık AKTAŞ	Memur	1750	Toki Konutları
5	Sıddık EKTİ	İşçi	1000	Emek Mah.

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

Görüldüğü üzere bilgileri veri tabanımıza doğru bir şekilde kaydettik.

1.11. Tabloya Alan Ekleme

Var olan bir tabloya alan ekleme ihtiyacı olabilir. Bu gibi durumlarda ise **ALTER TABLE** komutunu kullanırız. Bu komutun kullanımını şu şekildedir:

```
ALTER TABLE tablo_adi ADD alan_adi alan_turu;
```

Örnek: *Velibilgi* tablomuza telefon alanı ekleyelim.

```
mysql> alter table velibilgi ADD telefon int;
Query OK, 5 rows affected (0.12 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from velibilgi;
+----+-----+-----+-----+-----+-----+
| id | isim      | isi      | gelir | adres      | telefon |
+----+-----+-----+-----+-----+-----+
| 1  | Hasan OKUR | öğretmen | 1900  | Uarlık Mah. | NULL    |
| 2  | Hasan ASLAN | işçi     | 1500  | Emek Mah.   | NULL    |
| 3  | Recep KOÇAK | öğretmen | 1850  | Ekici Mah.  | NULL    |
| 4  | Sadık AKTAŞ | Memur    | 1750  | Toki Konutları | NULL    |
| 5  | Sıddık EKTİ | işçi     | 1000  | Emek Mah   | NULL    |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Burada telefon bilgilerini girmediğimiz için *NULL* değeri gözükmektedir.

1.12. Tablodan Alan Silme

Var olan tablodan bir sütünü silmek için ise şöyle bir ifade yazarız:

```
ALTER TABLE tablo_adi DROP COLUMN sütun_adi;
```

Örnek: *Velibilgi* tablomuzdan telefon alanını silelim.

```
mysql> select * from velibilgi;
+----+-----+-----+-----+-----+-----+
| id | isim      | isi      | gelir | adres      | telefon |
+----+-----+-----+-----+-----+-----+
| 1  | Hasan OKUR | öğretmen | 1900  | Uarlık Mah. | NULL    |
| 2  | Hasan ASLAN | işçi     | 1500  | Emek Mah.   | NULL    |
| 3  | Recep KOÇAK | öğretmen | 1850  | Ekici Mah.  | NULL    |
| 4  | Sadık AKTAŞ | Memur    | 1750  | Toki Konutları | NULL    |
| 5  | Sıddık EKTİ | işçi     | 1000  | Emek Mah   | NULL    |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> alter table velibilgi drop column telefon;
Query OK, 5 rows affected (0.05 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from velibilgi;
+----+-----+-----+-----+-----+
| id | isim      | isi      | gelir | adres      |
+----+-----+-----+-----+-----+
| 1  | Hasan OKUR | öğretmen | 1900  | Uarlık Mah. |
| 2  | Hasan ASLAN | işçi     | 1500  | Emek Mah.   |
| 3  | Recep KOÇAK | öğretmen | 1850  | Ekici Mah.  |
| 4  | Sadık AKTAŞ | Memur    | 1750  | Toki Konutları |
| 5  | Sıddık EKTİ | işçi     | 1000  | Emek Mah   |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Yukarıda da görüldüğü gibi alanımızı başarılı bir şekilde temizlemiş olduk.

1.13. Tablo Adını Değiştirme

Bir veri tabanında tablo adını değiştirmek için aşağıdaki şekilde bir yol izlenir:

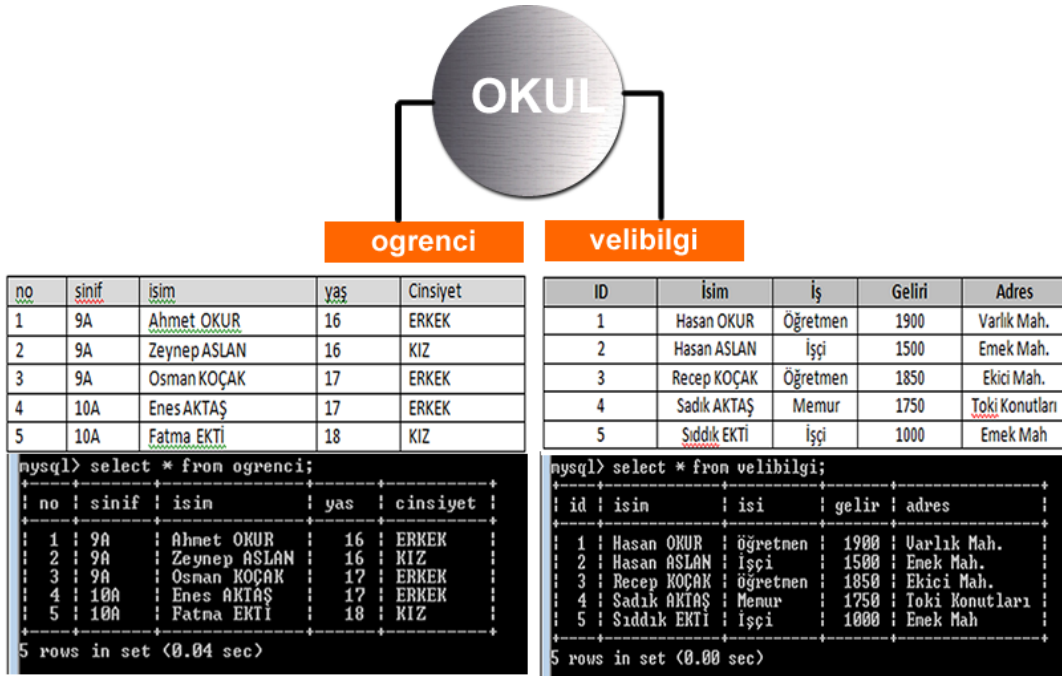
```
ALTER TABLE mevcut_tablo_adi RENAME TO yeni_tablo_adi;
```

```
mysql> alter table velibilgi rename velileri;  
Query OK, 0 rows affected (0.05 sec)
```

Komutu çalıştırdığımızda sıkıntısız bir şekilde tablo adı değişti.

Biz uygulamalarımızda *velibilgi* tablo adını kullanacağımız için eski ismine geri getiriyoruz.

Buraya kadar yaptığımız uygulamaları bir toparlayacak olursak *OKUL* adında bir veri tabanı oluşturmuştuk. Bunun içine “*ogrenci*” ve “*velibilgi*” adında iki tane tablo ekleyip tablolara ayrı ayrı veri girişi yaptık.



1.14. Sorgulama Operatörleri

Bazı durumlarda girdiğimiz verileri belli bir silsile dâhilinde çağırmanız gerekebilir. Bu gibi durumlarda sorgulama operatörlerini kullanırız.

Daha önce oluşturduğumuz örnekler üzerinde sorgulama işlemini gerçekleştirelim.

SELECT seçme manası gelir, **WHERE** ise nereden alınacağını söyler.

Bu komutun kullanımı şu şekildedir:

```
SELECT sütunlar FROM tablo_ismi WHERE koşullar;
```

Örnek: Velilere toplantı davetiyesi gönderilecek, veli isimleri ve adresleri gerekli, şimdi “*velibilgi*” tablomuzdan sadece velilerin ismini ve adresini çağıralım:

```
mysql> use okul
Database changed
mysql> select isim, adres FROM velibilgi;
+-----+-----+
| isim      | adres      |
+-----+-----+
| Hasan OKUR | Uurluk Mah. |
| Hasan ASLAN | Emek Mah.  |
| Recep KOÇAK | Ekici Mah.  |
| Sadık AKTAŞ | Toki Konutları |
| Sıddık EKTİ | Emek Mah   |
+-----+-----+
5 rows in set (0.00 sec)
```

Örnek: Öğrencilere sosyal yardımda bulunulacak. Velisinin geliri aylık 1.250 TL'nin altında olan veli listesini çıkartalım:

```
mysql> select isim, gelir FROM velibilgi WHERE gelir<=1250;
+-----+-----+
| isim      | gelir      |
+-----+-----+
| Sıddık EKTİ | 1000      |
+-----+-----+
1 row in set (0.06 sec)
```

Sorgu yaptırdığımızda bir kaydın çıktığı görünmektedir.

1.15. Koşula Bağlı Sorgulama Yapma

Yukarıdaki örnekte de uyguladığımız gibi *WHERE* parametresi koşula bağlı sorgulama işlemlerinde kullanılır. Bu parametrenin kullanımını birkaç örnekle biraz daha pekiştirelim.

Örnek: Cinsiyeti *KIZ* olan öğrencilerin listesini alalım:

```
mysql> use okul
Database changed
mysql> select * from ogrenci where cinsiyet="KIZ";
+----+-----+-----+-----+-----+
| no | sınıf | isim          | yas | cinsiyet |
+----+-----+-----+-----+-----+
| 2  | 9A    | Zeynep ASLAN | 16  | KIZ      |
| 5  | 10A   | Fatma EKTİ   | 18  | KIZ      |
+----+-----+-----+-----+-----+
2 rows in set (0.13 sec)
```

Örnek: Cinsiyeti *KIZ* olan *10A* sınıf öğrencilerinin listesini alalım:

```
mysql> select * from ogrenci where cinsiyet="KIZ" and sınıf="10A";
+----+-----+-----+-----+-----+
| no | sınıf | isim          | yas | cinsiyet |
+----+-----+-----+-----+-----+
| 5  | 10A   | Fatma EKTİ   | 18  | KIZ      |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- **Order By ASC/DESC:** Bu komut ile verileri alfabetik olarak sıralayabiliriz.
 - **ASC:** Artan-Küçükten büyüğe doğru (A-Z)
 - **DESC:** Azalan-Büyükten küçüğe doğru (Z-A) sıralama işlemi gerçekleştirir.

Örnek: Tüm öğrencileri alfabetik olarak sıralayalım (A-Z):

```
mysql> select * from ogrenci ORDER BY isim ASC;
+----+-----+-----+-----+-----+
| no | sınıf | isim          | yas | cinsiyet |
+----+-----+-----+-----+-----+
| 1  | 9A    | Ahmet OKUR   | 16  | ERKEK    |
| 4  | 10A   | Enes AKTAŞ   | 17  | ERKEK    |
| 5  | 10A   | Fatma EKTİ   | 18  | KIZ      |
| 3  | 9A    | Osman KOÇAK  | 17  | ERKEK    |
| 2  | 9A    | Zeynep ASLAN | 16  | KIZ      |
+----+-----+-----+-----+-----+
5 rows in set (0.04 sec)
```


Şimdi de isimleri tersten (Z'den-A'ya) göre sıralayalım. Bu sefer *ASC* yerine *DESC* ifadesini kullanıyoruz.

```
mysql> select * from ogrenci ORDER BY isim DESC;
+----+-----+-----+-----+-----+
| no | sinif | isim          | yas | cinsiyet |
+----+-----+-----+-----+-----+
| 2  | 9A    | Zeynep ASLAN | 16  | KIZ      |
| 3  | 9A    | Osman KOÇAK  | 17  | ERKEK    |
| 5  | 10A   | Fatma EKTİ   | 18  | KIZ      |
| 4  | 10A   | Enes AKTAŞ   | 17  | ERKEK    |
| 1  | 9A    | Ahmet OKUR   | 16  | ERKEK    |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

ASC-DESC komutu sadece metinsel ifadeleri değil sayısal ifadeleri de sıralamak için kullanılır.

Örneğin “velibilgi” tablosundan velilerin sosyal durumlarını öğrenmek için gelirlerine göre azdan çoğa doğru sıralatalım:

```
mysql> select * from velibilgi ORDER BY gelir ASC;
+----+-----+-----+-----+-----+
| id | isim          | isi          | gelir | adres          |
+----+-----+-----+-----+-----+
| 5  | Sıddık EKTİ  | İşçi        | 1000  | Emek Mah      |
| 2  | Hasan ASLAN  | İşçi        | 1500  | Emek Mah.     |
| 4  | Sadık AKTAŞ  | Memur       | 1750  | Toki Konutları |
| 3  | Recep KOÇAK  | öğretmen    | 1850  | Ekici Mah.    |
| 1  | Hasan OKUR   | öğretmen    | 1900  | Varlık Mah.  |
+----+-----+-----+-----+-----+
5 rows in set (0.03 sec)
```

- **Sorgu Sınırlama(LIMIT):** *LIMIT* parametresi ile sıraladığımız listeye bir sınır koyabiliriz. Şöyle ki geliri düşükten yükseğe doğru sadece 3 veli isminin listelenmesini istiyoruz.

```
mysql> select * from velibilgi ORDER BY gelir ASC LIMIT 3;
+----+-----+-----+-----+-----+
| id | isim          | isi          | gelir | adres          |
+----+-----+-----+-----+-----+
| 5  | Sıddık EKTİ  | İşçi        | 1000  | Emek Mah      |
| 2  | Hasan ASLAN  | İşçi        | 1500  | Emek Mah.     |
| 4  | Sadık AKTAŞ  | Memur       | 1750  | Toki Konutları |
+----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Şimdi de sıralama alanımızı biraz daraltıp *İşçi* statüsünde olan velilerin gelirlerini sıralayalım:

```
mysql> select * from velibilgi WHERE isi="İşçi" ORDER BY gelir ASC;
+----+-----+-----+-----+-----+
| id | isim      | isi   | gelir | adres  |
+----+-----+-----+-----+-----+
| 5  | Sıddık EKTİ | İşçi | 1000  | Emek Mah. |
| 2  | Hasan ASLAN | İşçi | 1500  | Emek Mah. |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

1.16. Verileri Güncelleme

Var olan veri yerine başka bir veri girmek için bu komut kullanılır.

Update: Verileri güncellemek için bu komut kullanılır.

Kullanımı şu şekildedir:

```
UPDATE tablo_ismi SET alan='Yeni Değer' where 'Değişecek Kayıt';
```

Örnek: Aşağıdaki tabloda 3 numaralı öğrenci Osman KOÇAK'ın isim bilgilerini güncelleyelim. KOÇAK yerine KOÇ, sınıfını da 9A yerine 10A olarak ayarlayalım:

```
mysql> select * from ogrenci;
+----+-----+-----+-----+-----+
| no | sinif | isim          | yas | cinsiyet |
+----+-----+-----+-----+-----+
| 1  | 9A    | Ahmet OKUR   | 16  | ERKEK    |
| 2  | 9A    | Zeynep ASLAN | 16  | KIZ      |
| 3  | 9A    | Osman KOÇAK  | 17  | ERKEK    |
| 4  | 10A   | Enes AKIŞ    | 17  | ERKEK    |
| 5  | 10A   | Fatma EKTİ   | 18  | KIZ      |
+----+-----+-----+-----+-----+
5 rows in set (0.04 sec)
```

Şimdi komutun uygulanış biçimine bir göz atalım:

```
mysql> update ogrenci set isim='Osman KOÇ', sınıf='10A' where no=3;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from ogrenci;
+----+-----+-----+-----+-----+
| no | sınıf | isim      | yas  | cinsiyet |
+----+-----+-----+-----+-----+
| 1  | 9A    | Ahmet OKUR | 16   | ERKEK    |
| 2  | 9A    | Zeynep ASLAN | 16   | KIZ      |
| 3  | 10A   | Osman KOÇ  | 17   | ERKEK    |
| 4  | 10A   | Enes AKTAŞ | 17   | ERKEK    |
| 5  | 10A   | Fatma EKTİ | 18   | KIZ      |
+----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

1.17. Hazır Fonksiyonlar

Sql dilinde kullanılan hazır fonksiyonlar vardır. Bunlar işlerimizi kolaylaştırmakla beraber sorguda bize kolaylık sağlar.

- **Count:** Toplam *Kayıt* sayısını bulur.

Kullanımı: **Count(Sütün_Adı)**

Örnek: Daha önce oluşturduğumuz *ogrenci* tablosundaki kayıt sayısını bulalım:

```
mysql> select count(no) FROM ogrenci;
+-----+
| count(no) |
+-----+
| 5          |
+-----+
1 row in set (0.05 sec)
```

Görüldüğü gibi tablomuzda 5 kayıt bulunmuş ve listelenmiştir. Sütun olarak ilk sütundaki verileri saydıracağız diye bir şart yoktur. İstenirse sınıf, isim, yaş, cinsiyet de saydırılabilir.

```
mysql> select count(isim) FROM ogrenci;
+-----+
| count(isim) |
+-----+
| 5           |
+-----+
1 row in set (0.00 sec)
```

Sonuç yine aynı çıkacaktır.

- **Min /Max:** Minimum ve maksimum değerleri bulur.

Örnek: *Ogrenci* tablosundaki öğrencilerden en küçük yaşı bulalım:

```
mysql> select min(yas) FROM ogrenci;
+-----+
| min(yas) |
+-----+
|        16 |
+-----+
1 row in set (0.04 sec)
```

Öğrencilerden en küçük yaş 16 bulundu.

Örnek: *Ogrenci* tablosundaki öğrencilerden en büyük yaşı bulalım:

```
mysql> select max(yas) FROM ogrenci;
+-----+
| max(yas) |
+-----+
|        18 |
+-----+
1 row in set (0.00 sec)
```

Öğrencilerden en yüksek yaş 18 bulundu.

- **Distinct:** Bu fonksiyon tabloda sütun içerisindeki benzer olan verilerden sadece bir tanesini sorguya ekler.

Bu fonksiyon için velibilgi tablomuzu kullanalım:

ID	İsim	İşi	Geliri	Adres
1	Hasan OKUR	Öğretmen	1900	Varlık Mah.
2	Hasan ASLAN	İşçi	1500	Emek Mah.
3	Recep KOÇAK	Öğretmen	1850	Ekici Mah.
4	Sadık AKTAŞ	Memur	1750	Toki Konutları
5	Sıddık EKTİ	İşçi	1000	Emek Mah.

Yukarıdaki tabloyu incelediğimizde 2 öğretmen, 2 işçi ve 1 memur var. Bu fonksiyon, bu bilgileri filtreleyip yalnız bir tanesini işleme sokacaktır.

Örnek: Velilerin hangi işlerle meşgul olduğunu görmek istersek

```
mysql> select distinct isi FROM velibilgi;
+-----+
| isi   |
+-----+
| Öğretmen |
| İşçi   |
| Memur  |
+-----+
3 rows in set (0.05 sec)
```

Dikkat ettiğiniz gibi bu fonksiyon tekrarlı verileri ayıklayıp birer numunelerini listelemektedir. Bu fonksiyonu elektronik tablolar programındaki veri filtreleme özelliğine de benzetebiliriz.

SQL komut ve fonksiyonları, buraya kadar anlattıklarımızla sınırlı olmayıp daha farklı türlerde olanları mevcuttur.

UYGULAMA FAALİYETİ

Veri tabanı üzerinde aşağıdaki işlemleri uygulayınız.

İşlem Basamakları	Öneriler
➤ Bir tane veri tabanı oluşturunuz.	➤ Create databases komutunu kullanınız.
➤ Veri tabanına ait iki tane tablo oluşturunuz.	➤ Create table komutunu kullanınız.
➤ Tabloların isimlerini değiştiriniz.	➤ Alter Table –Rename komutunu kullanınız.
➤ Tablolardan bir tanesini siliniz.	➤ Delete komutunu kullanınız.
➤ Oluşturduğunuz alanlara veri girişi yapınız.	➤ Insert Into komutunu kullanınız.
➤ Girilen verileri değiştiriniz.	➤ Update-Set komutunu kullanınız.
➤ Verileri farklı özelliklere göre sıralayınız.	➤ Select-Update komutunu kullanınız.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Sql komut sisteminde kullanılan show databases; komutunun görevi nedir?
A) Sunucu adını gösterir. B) Veri tabanlarını gösterir.
C) Tabloları gösterir. D) Veri tabanı biçimlerini gösterir.
2. Sql komut sisteminde kullanılan Create database; komutu hangi işlevi yerine getirir?
A) Veri tabanı oluşturur. B) Tablo oluşturur.
C) Veri oluşturur. D) Hiçbiri
3. Sql komut sisteminde kullanılan create table yazarlar; komutunun görevi nedir?
A) Yazarlar adında veri tabanı oluşturur.
B) Yazarlar adında tablo oluşturur.
C) Yazarlar adında alan oluşturur.
D) Yazarlar adında bilgi girişi yapar.
4. Delete From Musteri Where no=2; komutu Sql dilindeki karşılığı nedir?
A) Musteri tablosundan kayıt gösterir.
B) Musteri tablosundaki kayıtları siler.
C) Musteri tablosundaki 2 numaralı kayıdı siler.
D) Musteri tablosunun no alanına 2 değerini atar.
5. “Auto increment primary key” Sql dilinde hangi manaya gelir?
A) Otomatik artan farkı değer B) Sıralı ikili dizisi
C) Değiştirilemeyen alan D) Apache
6. Sql komut sisteminde kullanılan Select isim, email FROM uyeler”; komutunun görevi nedir?
A) Uyeler tablosuna isim ve email alanı açar.
B) Uyeler tablosundan isim ve email alanını çıkarır.
C) Uyeler tablosundaki isim ve email bilgileri haricindeki listeler.
D) Uyeler tablosundaki isim ve email bilgilerini listeler.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Veri tabanı programlamasını yapabileceksiniz.

ARAŞTIRMA

- Veri tabanı işlevlerini kolaylaştıran ve görsel arayüz sağlayarak işlemlerimizi kolaylaştıran yazılımlar nelerdir? Araştırmız.

2. PHP İLE MYSQL PROGRAMLAMA

Buraya kadar gördüğümüz üzere SQL dilini DOS komut sistemi ekranına benzer bir ekranda kodlayıp, veriler girip düzeltmek çok kolay bir iş değildir. Bunun için piyasada çeşitli programlar mevcuttur. Bunların en bilinenleri MYSQL-Front ve PhpMyAdmin programlarıdır. Bu programlar, görsel olarak bizim yaptığımız işlevleri arka planda SQL komutlarına çevirir.

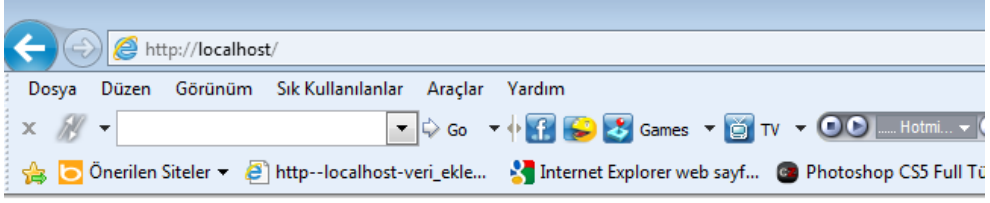
Bu uygulama faaliyeti içerisinde PHP ile uyumlu çalışan PhpMyAdmin programını inceleyeceğiz.

2.1. Veri Tabanı Programlarıyla Çalışma

PhpMyAdmin, PHP web programlama dili ile kodlanmış olan ve MySQL veri tabanı yönetim desteği sağlayan bir web uygulamasıdır.

Dersimizin ilk modülünde yer alan konular içerisinde kurulumunu yapmış olduğumuz PHP+Apache+Mysql paketleri içinde yer alan PhpMyAdmin'den faydalanacağız.

Öncelikle PhpMyAdmin programımızı çalıştırmak için web tarayıcımızın adres çubuğuna <http://localhost> ya da 127.0.0.1 yazıyoruz. Karşımıza gelen pencereden phpMyAdmin Database Manager seçeneğini seçiyoruz.



The AppServ Open Project - 2.5.10 for Windows

phpMyAdmin Database Manager Version 2.10.3
PHP Information Version 5.2.6

About AppServ Version 2.5.10 for Windows

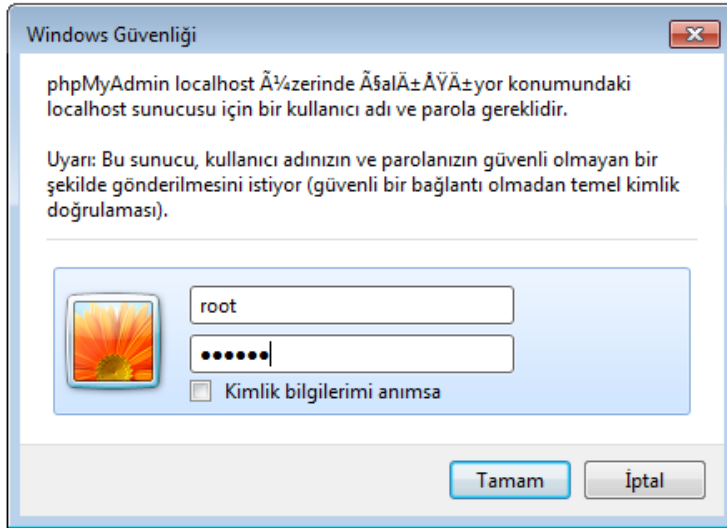
AppServ is a merging open source software installer package for Windows includes :

- Apache Web Server Version 2.2.8
- PHP Script Language Version 5.2.6
- MySQL Database Version 5.0.51b
- phpMyAdmin Database Manager Version 2.10.3

Resim 2.1: Appserv bilgi ekranı

2.2. Yeni Bir Veri Tabanı Oluşturma

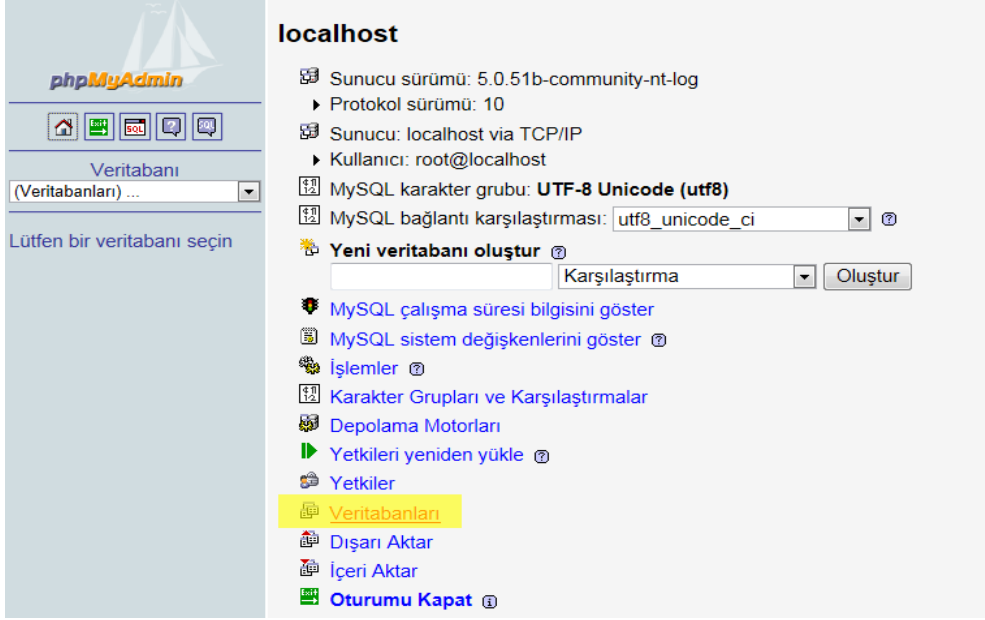
Burada veri tabanı sunucumuza bağlanıp veri tabanı işlemlerini gerçekleştireceğiz. Bazı durumlarda yukarıdaki ekran gelmeyebilir. PhpMyAdmin'e bağlanmak için *http://127.0.0.1/phpMyAdmin/* veya *http://localhost/phpMyAdmin* ifadelerini adres çubuğuna girmemiz gerekebilir.



Resim 2.2: PhpMyAdmin ekranına giriş paneli

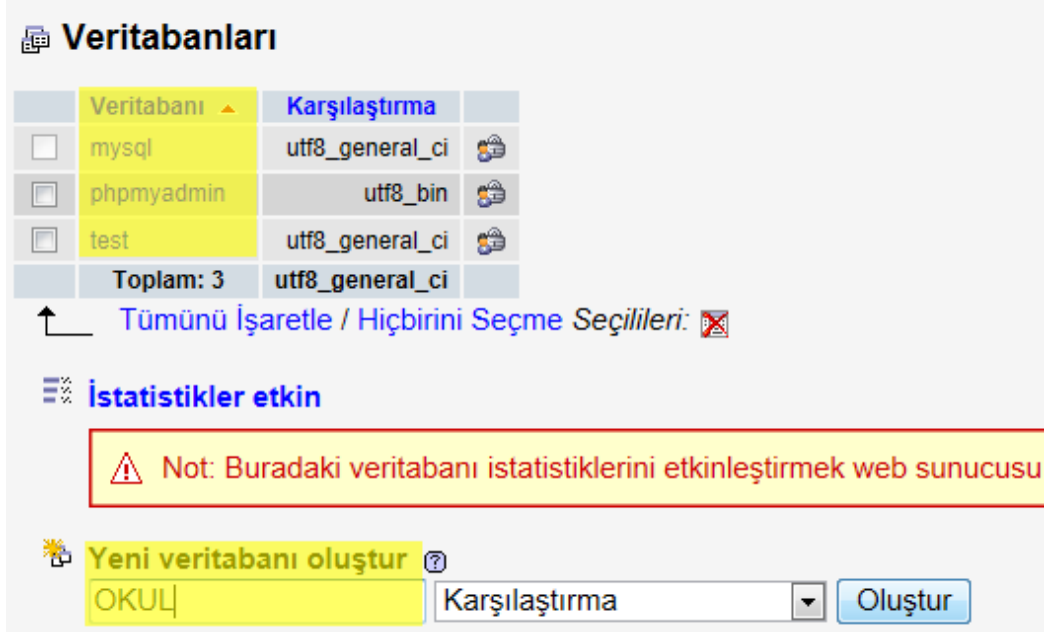
Gelen erişim ekranının üst kısmına kullanıcı adımızı, altına ise kurulumda kullandığımız şifreyi giriyoruz. Kullanıcı adı, kurulumda varsayılan (default) olarak root ayarlanmıştır. Parola ise bizim girdiğimiz değer olacaktır.

Görülen bu ekran bizim veri tabanı yönetim ekranımız olacaktır.



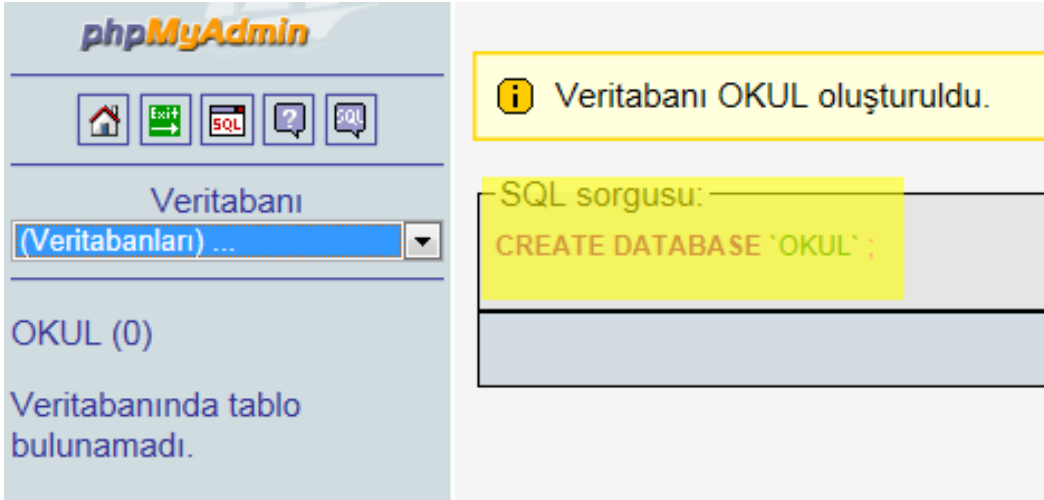
Resim 2.3: PhpMyAdmin ekranı

Sol tarafta veri tabanlarının listelemesi yapılabilmektedir. Veri tabanları yazan bağlantıya tıklayarak mevcut veri tabanlarının listesi görülebilir ve yeni veri tabanları oluşturulabiliriz.



Resim 2.4: PhpMyAdmin ekranından veri tabanları görüntüsü

Yukarıdaki pencereden faydalanarak sunucuda 3 tane veri tabanı olduğunu görüyoruz. Biz de OKUL adında yeni bir veri tabanı oluşturalım. Veri tabanının adını yazıp “Oluştur” diyoruz.




Resim 2.5: PhpMyAdmin ekranında veri tabanı oluşturma

Veri tabanı oluşturuldu. Aynı zamanda veri tabanı oluşturmak için kullanılan SQL komutunun sorgusuna da ekranda görebilirsiniz.

Şimdi de oluşturduğumuz veri tabanı içine tablo ekleme aşamasına sıra geldi.

Veritabanında tablo bulunamadı.

 OKUL veritabanında yeni tablo oluşturun

İsim:

Alan sayısı:

Resim 2.6: PhpMyAdmin ekranında alan oluşturma

Veri tabanımızda kullanacağımız tablo ismi *ogrencibilgi* olsun. Bu tablo da toplam 5 alandan oluşsun. *Git* butonuna tıkladığımızda karşımıza aşağıdaki şekilde bir tablo gelecektir:

Sunucu: localhost Veritabanı: OKUL Tablo: ogrencibilgi

Alan	Türü	Uzunluk/Değerler ¹	Karşılaştırma	Öznelikler	Boş	Varsayılan ²	Ekstra				
	VARCHAR				not null						
	VARCHAR				not null						
	VARCHAR				not null						
	VARCHAR				not null						
	VARCHAR				not null						

Tablo yorumları: Depolama Motoru: MyISAM Karşılaştırma:

Resim 2.7: PhpMyAdmin ekranında alan biçimi

Gelen bu tabloyu aşağıdaki gibi dolduralım:

Sunucu: localhost Veritabanı: okul Tablo: ogrencibilgi

Alan	Türü	Uzunluk/Değerler ¹	Karşılaştırma	Öznelikler	Boş	Varsayılan ²	Ekstra				
id	INT	4			not null		auto_increment				
isim	TEXT	25			not null						
sinif	VARCHAR	5			not null						
alan	VARCHAR	30			not null						
email	VARCHAR	40			not null						

Resim 2.8: PhpMyAdmin ekranında alan biçimi belirleme

Burada *id* (identification – Kimlik) alanını öğrenci numarası gibi düşünebiliriz. *No* yerine *id* kullandık. *Id* aynı değeri almayan ve otomatik değer alan tam sayı türünde olan ifadeler için kullanacağımız alan adıdır. Bunun için tablomuzun *Ekstra* bölümünde *id* alanına *auto_increment* (otomatik değer alma) özelliği vermiş olduk.

Diğer alanları da *text* ve *varchar* karakter olacak şekilde ayarlayıp alabileceği karakter değeri olarak da belli uzunluklar tahsis ettik. Bu değerler yukarıdaki tablomuzda görülmektedir.

2.3. Veri Tabanı Sunucusuna Bağlantı Kurma

Veri tabanı sunucumuza bağlanma komutu olarak **mysql_connect()** fonksiyonu kullanılır.

Bu fonksiyonun kullanımı aşağıdaki gibidir:

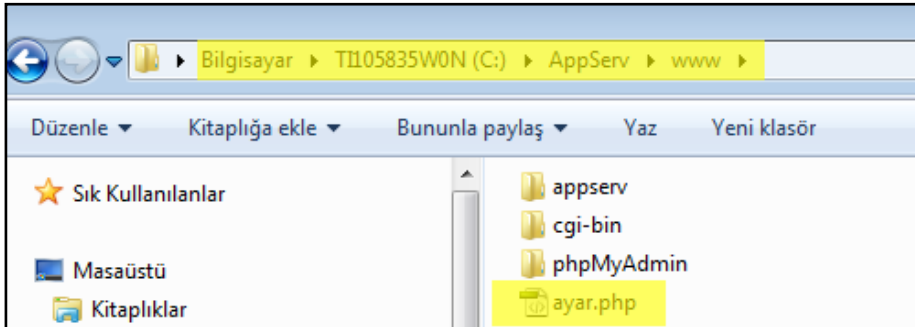
mysql_connect(sunucu_adi, kullanıcı_adi, sifre)

Kullanıcı adı ve parola veri tabanı programı kurma esnasında girdiğimiz değerlerdir. Bu fonksiyon veri tabanı sunucumuza bağlanma komutudur. MySQL sunucusu ile bağlantı kuramazsa **FALSE** değerini döndürür. İstenirse döndürdüğü bu değer, bir kontrol yapısı içinde kullanılarak bağlantının yapıp yapılmadığı kontrol edilebilir.

İlk olarak `ayar.php` dosyamızı oluşturup programımıza, Mysql sunucumuza tanıtalım.

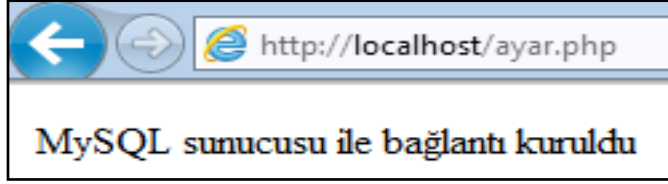
```
1 <?
2 $host="localhost"; //Sunucu adı
3 $kadi="root"; //Veri tabanı kullanıcı adı
4 $sifre="000000"; //Veri tabanı kullanıcı şifresi
5 $veritabani="okul"; //Veri tabanı adı
6 $tablo="ogrencibilgi";
7
8 $baglanti=mysql_connect($host,$kadi,$sifre); //Sunucuya bağ.kuruluyor
9 if($baglanti){ //Bağlantı olup olmadığını kontrol ediyor
10 echo "MySQL sunucusu ile bağlantı kuruldu";
11 }else{//Bağlantı kurulamamışsa
12 echo "MySQL sunucusu ile bağlantı kurulamadı. Hata kodu : ".
mysql_error();//Bağlantı hatasa veriyor
13 }
14 @mysql_select_db($veritabani,$baglanti)//Veri tabanını seçtik
15 or die ("Veri tabanına bağlanamadık");|
16 ?>
```

Oluşturduğumuz `ayar` dosyamızı sunucumuzun bulunduğu dizinde `www` klasörü içine *ayar.php* olarak kaydedelim.



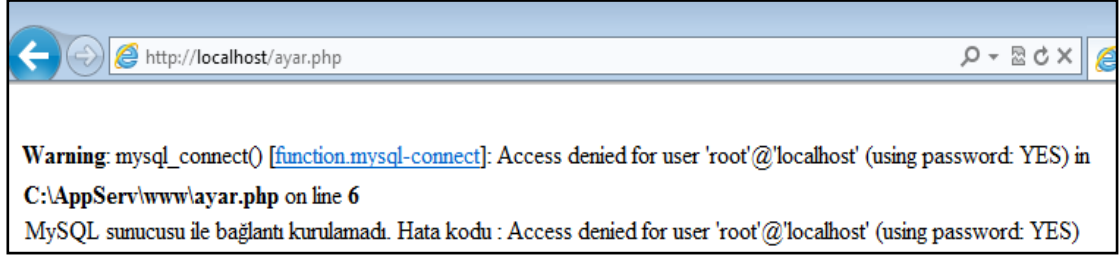
Resim 2.9: `ayar.php` dosyasının kayıt konumu

Tarayıcımızın adres çubuğuna ilgili adresi yazıp çalıştırdığımızda ise karşımıza şu şekilde bir ekran gelecektir:



Resim 2.10: Ayar.php dosyasını test etme -1

Buradan yaptığımız bağlantının doğru olduğu anlıyor ve bir sonraki aşamaya geçebileceğimiz sonucuna varıyoruz. Ayar.php dosyamızda 2, 3 ve 4 numaralı satırların değerlerinden herhangi birini değiştirdiğimiz zaman bağlantı hatası alabiliriz. Bu değerlerden birini değiştirerek ayar.php dosyamızı tekrar çalıştıralım.



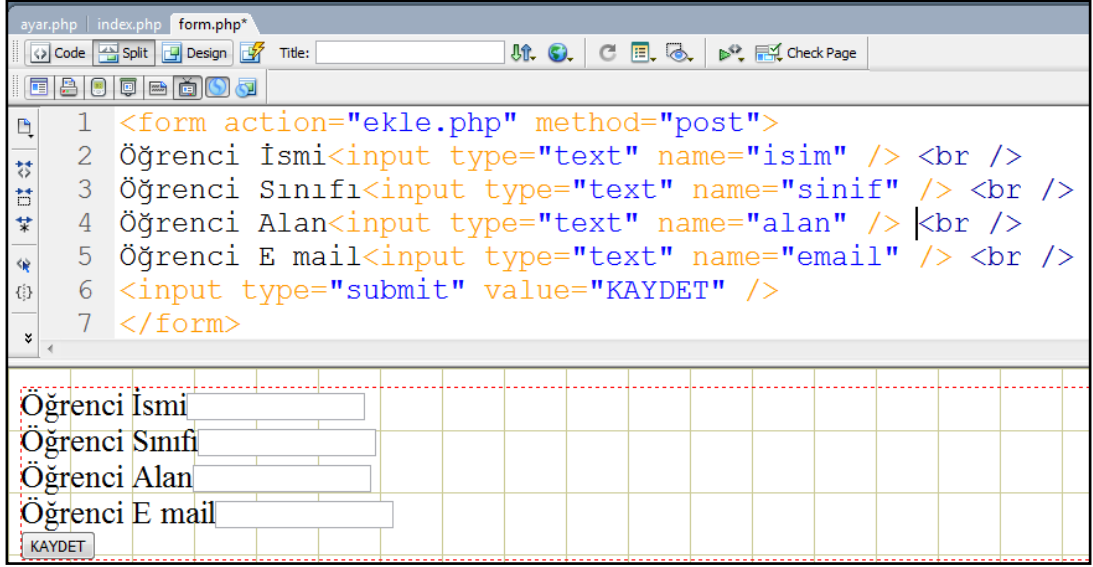
Resim 2.11: Ayar.php dosyasını test etme -2

Yanlış şifre değeri girildiğinde alınan hata yukarıdaki gibi olacaktır.

Veri tabanımıza bağlantıyı sağladıktan sonra veri tabanımıza veriler girmeye sıra geldi. Öncelikle PhpMyAdmin programı ile daha önce oluşturduğumuz veri tabanını ve tabloyu hatırlayalım.

id	isim	sınıf	alan	email
İnt(4)	Text(25)	Varchar(5)	Varchar(30)	Varchar(40)

Veri ekleme işlemini daha hızlı gerçekleştirmek için form kullanalım. Web editörü programı kullanarak bu işi kolayca yapabiliriz.



Form.php dosyamızı kaydettik. Şimdi de form ile gönderilen bilgileri veri tabanına eklemeye sıra geldi. Bunun için de *Ekle.php* dosyamızı oluşturalım.

2.4. Tabloya Bilgi Ekleme

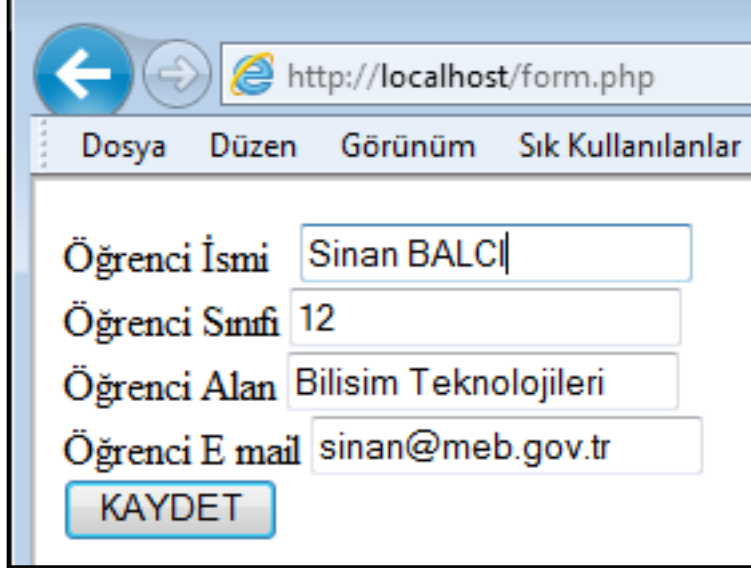
Form ile gönderilen bilgiler *\$_POST* metodu ile alınmaktadır. Aldığımız bu bilgileri 3-6 satırlar arasında değişkenlere atıp 12. satırda veri tabanına ekliyoruz.

```
1 <?
2 include("ayar.php");//ayarlarımızı sayfaya çağırıyoruz
3 $isim=$_POST["isim"]; //Forumdan gelen ismi, isim değişkene atadık.
4 $sinif=$_POST["sinif"]; //Forumdan gelen sınıfı, sınıf değişkene atadık.
5 $alan=$_POST["alan"]; //Forumdan gelen alanı, alan değişkene atadık.
6 $email=$_POST["email"]; //Forumdan gelen emaili, email değişkene atadık.
7 echo $isim ; echo "<br>"; //Değerleri önce ekrana yazdırıyoruz
8 echo $sinif; echo "<br>";
9 echo $alan; echo "<br>";
10 echo $email; echo "<br>";
11
12 $ekle=mysql_query("insert INTO $tablo(isim,sinif,alan,email)
13 values('$isim','$sinif','$alan','$email')");
14 if($ekle) { //Ekleme işlemi başarılıysa
15 echo "Ekleme işlemi başarılı <br> YENİ KAYIT İÇİN <a href=ekle.php>
16 TIKLAYIN</a>"; } //Yeni kayıt için ekle.php ye geri dönüyoruz
17 else { //Ekleme işlemi başarısızsa
18 echo "Ekleme işlemi Başarısız"; }
19 ?>
```

12. satırda ekleme işlemi \$ekle değişkeni ile kontrol edilmektedir. *Insert INTO* komutu ise ekleme işlemi gerçekleştirilmesini sağlar.

Sayfamızı ekle.php olarak kaydederek buraya kadar kodları çalıştıralım.

Adres çubuğuna `http://localhost/form.php` yazıp formumuzu çağırıyoruz.



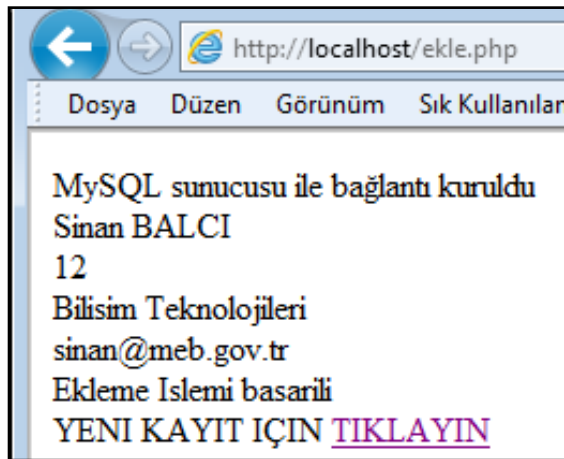
The screenshot shows a web browser window with the address bar containing `http://localhost/form.php`. The browser's menu bar includes 'Dosya', 'Düzen', 'Görünüm', and 'Sık Kullanılanlar'. The main content area displays a registration form with the following fields and values:

Öğrenci İsmi	Sinan BALCI
Öğrenci Sınıfı	12
Öğrenci Alan	Bilisim Teknolojileri
Öğrenci E mail	sinan@meb.gov.tr

Below the form is a button labeled 'KAYDET'.

Resim 2.12: Form.php dosyasını test etme

Formumuzu doldurup *KAYDET* butonuna tıklıyoruz. Kaydet tuşundan sonra ekle.php dosyası çağrılacak ve forma girdiğimiz bilgiler ekle.php yardımıyla veri tabanımıza kaydedilecektir.



The screenshot shows a web browser window with the address bar containing `http://localhost/ekle.php`. The browser's menu bar includes 'Dosya', 'Düzen', 'Görünüm', and 'Sık Kullanılanlar'. The main content area displays the following text:

MySQL sunucusu ile bağlantı kuruldu
Sinan BALCI
12
Bilisim Teknolojileri
sinan@meb.gov.tr
Ekleme İşlemi başarılı
YENİ KAYIT İÇİN [TIKLAYIN](#)

Resim 2.13: Kayıt ekleme

Veri tabanımıza farklı üç tane daha kayıt girmek için forma geri dönelim. Toplam 4 adet kayıt girmiş olduk ve şimdi tarayıcımızın adres çubuğuna <http://localhost/PhpMyAdmin> adresini yazıp veri tabanına bağlanalım.



Resim 2.14: PhpMyAdmin penceresi

PhpMyAdmin penceresinde *Gözet* butonuna tıklayarak kaydettiğimiz verilere ulaşabiliyoruz.



Resim 2.14: PhpMyAdmin penceresinde kayıt listeleme

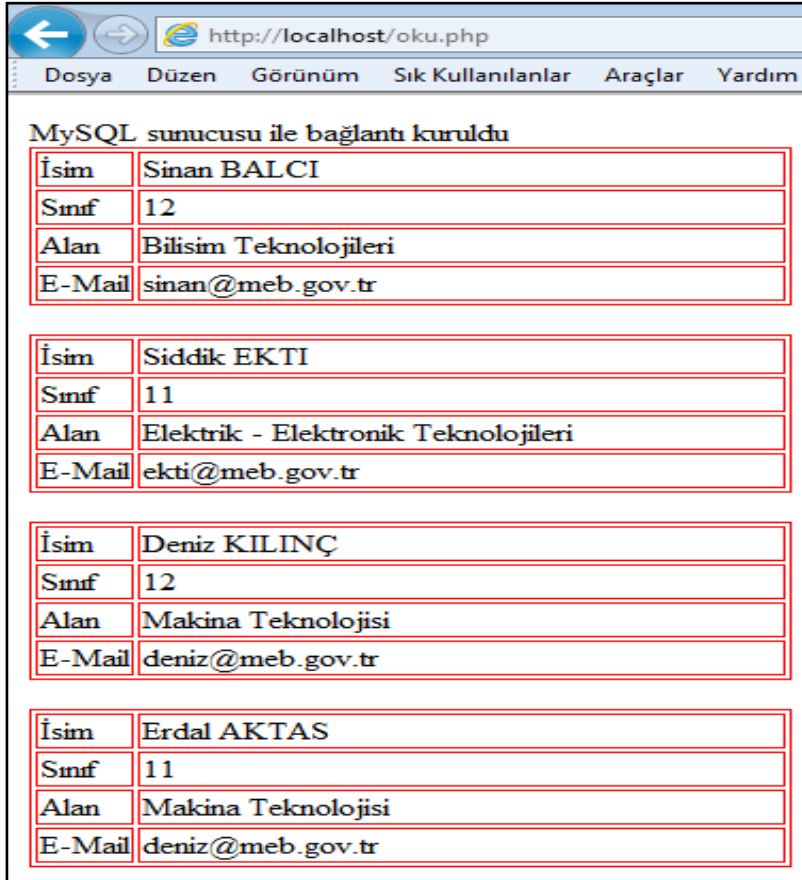
2.5. Tablodan Bilgileri Okutma

Veri tabanına girdiğimiz bu verileri okutmaya sıra geldi. Bunun için *oku.php* dosyasını oluşturuyoruz.

```
1 <?php
2 include("ayar.php");
3 $sor=mysql_query("select * from ogrencibilgi");//Kayıt sayısı
4 while($cek=mysql_fetch_array($sor)) {
5 //Tüm kayıtlar dizi içine atılıyor
6 $id=$cek["id"];
7 $isim=$cek["isim"];
8 $sinif=$cek["sinif"];
9 $alan=$cek["alan"];
10 $email=$cek["email"];
11 ?>
12 <table border="1" bordercolor="#FF0000">
13 <tr><td>İsim</td> <td width="300"><? echo $isim ?></td></tr>
14 <tr><td>Sınıf</td> <td width="300"><? echo $sinif ?></td></tr>
15 <tr><td>Alan</td> <td width="300"><? echo $alan ?></td></tr>
16 <tr><td>E-Mail</td> <td width="300"><? echo $email ?></td></tr>
17 </table> <br />
18 <?
19 }
20 ?>
```

Yukarıdaki örneği inceleyecek olursak 3. satırda veri tabanımıza kayıtlı bilgi adedi *\$sor* değişkenine atılıyor. 4. satırda ise *\$sor* değişkenini *mysql_fetch_array* fonksiyonu yardımı ile sorgudan dönen değişkeni diziye atmak için kullanılıyor. Kurduğumuz *while* döngüsü kayıt bitinceye kadar devam etmektedir. 6 ile 10. satır arasında dizi değerleri sırası ile değişkenlere atılmaktadır. 12 ile 17. satırda ise tablo yardımıyla veri tabanından çekilen değerler ekrana yazılmaktadır. Yazma işlemi döngü bitinceye kadar devam etmektedir.

Oku.php dosyasını çalıştırdığımızda ekrana aşağıdaki gibi bir görüntü gelecektir:



MySQL sunucusu ile bağlantı kuruldu	
İsim	Sinan BALCI
Sınıf	12
Alan	Bilisim Teknolojileri
E-Mail	sinan@meb.gov.tr
İsim	Siddik EKTI
Sınıf	11
Alan	Elektrik - Elektronik Teknolojileri
E-Mail	ekti@meb.gov.tr
İsim	Deniz KILINÇ
Sınıf	12
Alan	Makina Teknolojisi
E-Mail	deniz@meb.gov.tr
İsim	Erdal AKTAS
Sınıf	11
Alan	Makina Teknolojisi
E-Mail	deniz@meb.gov.tr

Resim 2.15: Verilerin listelenmesi (oku.php)

2.6. Tablodan Bilgileri Düzenleme

Girdiğimiz kayıtları düzenlemek için oku.php sayfamızda tablonun altına *DÜZENLE* adında bir bağlantı (link) oluşturuyoruz. Bu linke ait id parametresi düzenlenecek kaydın id numarası olup *Duzenle.php* ekranına ilgili sayfayı getirecektir.

```
18 <tr>
19 <td><a href="duzenle.php?id="<?php echo "$id"; ?>">DÜZENLE </a></td>
20 <td>&nbsp;</td>
21 </tr>
22 </table> <br />
23 <?
24 }
25 ?>
26
```

İsim	
Sınıf	
Alan	
E-Mail	
DÜZENLE	

Burada çağrılan id değeri href="duzenle.php?id="<?php echo "\$id";?> değeridir. Önce duzenle.php dosyamızı bir oluşturalım. Aşağıda duzenle.php dosyamız mevcuttur:

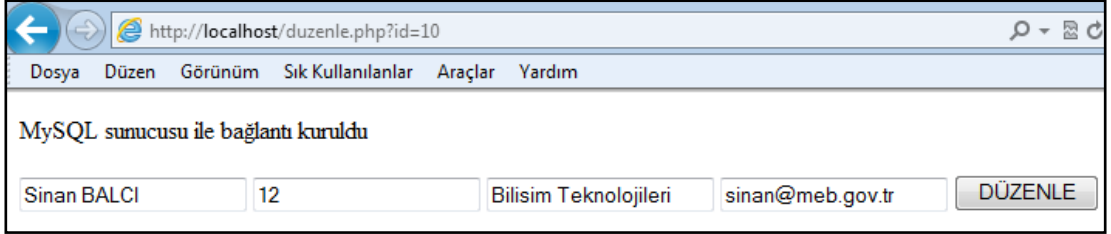
```
1 <?php
2 include("ayar.php");
3 $id=intval($_GET["id"]); //id değerini integer çevr.
4 //Gelen id değeri üzerinden aşağıdaki kayıt veri tabn. çekiliyor
5 $bilgial=mysql_fetch_array(mysql_query("select * from ogrencibilgi where id='$id'"));
6 $isim=$bilgial['isim'];
7 $sinif=$bilgial['sinif'];
8 $alan=$bilgial['alan'];
9 $email=$bilgial['email'];
10 ?>
11 <form action="duzenlemeyap.php?id="<? echo "$id"; ?>" method="post">
12 <input name="isim" value="<?php echo "$isim"; ?>" >
13 <input name="sinif" value="<?php echo "$sinif"; ?>" >
14 <input name="alan" value="<?php echo "$alan"; ?>" >
15 <input name="email" value="<?php echo "$email"; ?>" >
16 <input type="submit" value="DÜZENLE">
17 </form>
```

Tarayıcımızla "oku.php"yi çalıştırıp oradan da "DÜZENLE" linkine tıkladığımızda karşımıza düzenlenecek sayfa gelecektir.

MySQL sunucusu ile bağlantı kuruldu	
İsim	Sinan BALCI
Sınıf	12
Alan	Bilgisim Teknolojileri
E-Mail	sinan@meb.gov.tr
DÜZENLE	

Resim 2.16: Oku.php dosyası

Burada *DÜZENLE* linkine tıkladığımızda *id* değeri yardımıyla *duzenle.php* dosyasından ilgili kayıt çağrılacaktır.



MySQL sunucusu ile bağlantı kuruldu

<input type="text" value="Sinan BALCI"/>	<input type="text" value="12"/>	<input type="text" value="Bilisim Teknolojileri"/>	<input type="text" value="sinan@meb.gov.tr"/>	<input type="button" value="DÜZENLE"/>
--	---------------------------------	--	---	--

Resim 2.17: Duzenle.php dosyasını test etme

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
➤ Var olan veri tabanı sunucunuza bağlantı kurunuz.	➤ Ayar.php dosyası oluşturabilirsiniz.
➤ Bir adet veri tabanı oluşturunuz.	➤ PhpMyAdmin programını kullanınız.
➤ Var olan veri tabanına tablo ve alanlar ekleyiniz.	➤ PhpMyAdmin programını kullanınız.
➤ Alanlara php programlama dili yardımıyla bilgi girişi yapınız.	➤ Form oluşturarak POST metodu yardımıyla bu işi gerçekleştiriniz.
➤ Veri tabanına eklediğiniz bilgileri php programı yardımıyla okutunuz.	➤ While döngüsü yardımıyla veri tabanından verileri çekiniz.
➤ Mevcut bilgileri php programlama dili yardımıyla değiştiriniz.	➤ Update ve Set komutlarını kullanınız.
➤ Verilerinizden birkaç tanesini php programlama dili yardımıyla silmeyi deneyiniz.	➤ Delete ve Where komutlarını kullanınız.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi PhpMyAdmin için yanlıştır?
A) Görsel bir veri tabanı erişim arayüzüdür.
B) Php ile uyumlu çalışır.
C) Tüm veri tabanı programlarında kullanılabilir.
D) MySQL veri tabanları için hazırlanmış bir uygulamadır.
2. Mysql_connect (ABC, kullanıcı_adi, sifre) komut satırında ABC yerine yazılması gereken aşağıdakilerden hangisidir?
A) Veri tabanı adı
B) Alan adı
C) Tablo adı
D) Sunucu adı
3. @mysql_select_db(ABC,\$baglanti) komut satırında ABC yerine yazılması gereken aşağıdakilerden hangisidir?
A) Veri tabanı adı
B) Alan adı
C) Tablo adı
D) Sunucu adı
4. Ayar.php dosyası oluşturmanın mantığı nedir?
A) Her sayfa için ayrı ayrı bağlantı dosyası oluşturmaktan kurtarması
B) Sadece çağrıldığı dosyaları etkilemesi
C) Genelde include “ayar.php” şeklinde kullanılması
D) Hepsi
5.

```
3 $sor=mysql_query("select * from ogrencibilgi")
4 while($cek=mysql_fetch_array($sor)) {
```

Yukarıda görülen komut satırının görevi nedir?
A) Öğrenci tablosundan bilgi seçmek
B) Öğrenci tablosundaki bilgileri silmek
C) Kayıt sayısını sorgulamak
D) Öğrenci tablosundaki bilgiler adedince döngü oluşturup dizide kullanmak

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. MySql yazılımı ile ilgili aşağıda verilenlerden hangisi doğrudur?
A) Veri tabanı yazılımıdır.
B) Script (Betik) dilidir.
C) İşletim sistemidir.
D) Programlama dilidir.
2. Aşağıdakilerden hangisi veri tabanı ismi olabilir?
A) Okul Bilgileri
B) Okul_Bilgileri
C) \$Okul Bilgileri
D) Hiçbiri
3. Aşağıdakilerden hangisi veri tabanı oluşturmak için kullanılan SQL komutudur?
A) Create Table
B) Alter Table
C) Delete Database
D) Create Database
4. Aşağıdakilerden hangisi tablo oluşturmak için kullanılan SQL komutudur?
A) Alter Table
B) Delete Table
C) Create Table
D) Create Database
5. Aşağıdakilerden hangisi tablo içerisindeki bilgileri sorgulamak için kullanılan SQL komutudur?
A) SELECT
B) DELETE
C) INSERT
D) UPDATE
6. *Select Count(*) From tablo* komutunun görevi aşağıdakilerden hangisidir?
A) Tüm kayıtları listeler.
B) Tüm kayıtları siler.
C) Toplam kayıt sayısını verir.
D) Kayıtları A'dan Z'ye sıralar.

7. ALTER TABLE mevcut_tablo_adi RENAME TO yeni_tablo_adi; komutunun görevi aşağıdakilerden hangisidir?
A) Mevcut tablonun adını değiştirmeye yarar.
B) Mevcut veri tabanının adını değiştirmeye yarar.
C) Mevcut veri tabanının yapısını değiştirmeye yarar.
D) Mevcut tablonun yapısını değiştirmeye yarar.
8. ALTER TABLE tablo_adi DROP COLUMN sütun_adi; komutunun görevi aşağıdakilerden hangisidir?
A) Mevcut tabloyu siler.
B) Mevcut alanı siler.
C) Yeni alan ekler.
D) Mevcut kaydı siler.
9. DISTINCT fonksiyonunun görevi aşağıdakilerden hangisidir?
A) Tabloda sütun içlerindeki benzer olan verilerden sadece bir tanesini sorguya ekler.
B) Tabloda sütun içlerindeki verilerin toplamını verir.
C) Tabloda sütun içlerindeki verilerden en büyüğünü bulur.
D) Tabloda sütun içlerindeki verilerden en küçüğünü bulur.
10. Aşağıdakilerden hangisi bir veri tabanı türüdür?
A) MySQL
B) PHP
C) Apache
D) phpMyAdmin
11. phpMyAdmin ne tür bir yazılımdır?
A) Bir çeşit veri tabanıdır.
B) Script dilidir.
C) Mysql veri tabanına kolay erişim için hazırlanmış bir yazılımdır.
D) Web sunucusunu yapılandırmak için hazırlanmış bir yazılımdır.
12. mysql_connect() fonksiyonun işlevi aşağıdakilerden hangisidir?
A) Tabloları ilişkilendirmek için kullanılır
B) Veri tabanı sunucusuna bağlanmak için kullanılır.
C) Veri tabanı oluşturmak için kullanılır.
D) Veri tabanı sunucusuyla bağlantıyı kesmek için kullanılır.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	B
2	A
3	B
4	C
5	A
6	D

ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	C
2	D
3	A
4	D
5	D

MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1	A
2	B
3	D
4	C
5	A
6	C
7	D
8	B
9	A
10	A
11	C
12	B

KAYNAKÇA

- <http://www.php.net/>(12-03-2012/11:30)
- <http://www.mysql.com/>(16-03-2012/08:55)
- <http://www.easyphp.org/>(09-03-2012/15:10)
- ALTINKAYA Muhittin, Yahya DEMİRCAN, **MySQL ve Programlamaya Giriş**, Ankara, 2005.