

WEB TASARIM I

Öğr. Gör. M. Mutlu YAPICI

Ankara Üniversitesi
Elmadağ Meslek Yüksekokulu

Ders İzlenesi

Hafta	Modüller/İçerik/Konular
1. Hafta	PHP Tanımı ve Sunucu Kurulumları
2. Hafta	PHP Yazım Notasyonu ve Değişkenler
3. Hafta	PHP de Karar kontrol yapıları ve Döngüler
4. Hafta	Dizi ve Dizi işlemleri
5. Hafta	Fonksiyon, Sınıf ve Nesne Kavramları
6. Hafta	HTML ve PHP
7. Hafta	PHP ile Veritabanı İşlemleri
8. Hafta	MYSQL, MYSQLİ
9. Hafta	AJAX
10. Hafta	
11. Hafta	
12. Hafta	
13. Hafta	
14. Hafta	

Bu Ünite de Ele Alınan Konular

- HTML, CSS, JavaScript ve PHP
- PhpMyAdmin işlemleri
- Veritabanı bağlantı işlemleri
- Veritabanı seçme işlemleri
- Veritabanı sorgu işlemleri
- Veritabanı kayıt listeleme işlemleri
- Veritabanı kayıt ekleme işlemleri
- Veritabanı kayıt güncelleme işlemleri
- Veritabanı kayıt silme işlemleri
- Veritabanı bellek boşaltma işlemleri
- Veritabanı bağlantı kapatma işlemleri

Ders Kazanımları

Bu bölümü Bitirdiğimizde,

- HTML, CSS, JavaScript ve PHP
- PhpMyAdmin işlemleri
- Veritabanı bağlantı işlemleri
- Veritabanı seçme işlemleri
- Veritabanı sorgu işlemleri
- Veritabanı kayıt listeleme işlemleri
- Veritabanı kayıt ekleme işlemleri
- Veritabanı kayıt güncelleme işlemleri
- Veritabanı kayıt silme işlemleri
- Veritabanı bellek boşaltma işlemleri
- Veritabanı bağlantı kapatma işlemleri

öğrenmiş olacaksınız.

HTML, PHP ve MYSQL

PHP de veritabanı işlemleri için veritabanı MYSQL fonksiyonları kullanılmaktadır. PHP ile veritabanı işlemlerine geçmeden önce MYSQL veritabanında nasıl veritabanı oluşturabiliriz kısaca onu öğrenelim.

MYSQL veritabanı için oldukça kullanışlı olan ve kod yazmadan kolayca veritabanı oluşturmamızı tablo eklememizi ve alanlar yaratmamızı sağlayan ayrıca SQL sorgularını çalıştırmamızı sağlayan bir internet arayüzü bulunmaktadır. PhpMyAdmin dediğimiz bu arayüzle kolaylıkla veritabanı işlemi yapabilmekteyiz.

HTML, PHP ve MYSQL

Kurduğumuz sunucu içerisinde bu arayüze adres çubuğuna <http://localhost:8080/phpmyadmin/> yazarak ulaşabiliriz. Kullanım paneline bir kullanıcı adı ve bir de şifre ile giriş yapmaktayız. Varsayılan kullanıcı adı ve şifre panelin altında yazmaktadır.

Ayrıca bu kullanıcı adı ve şifreyi ileride veritabanına bağlanmak için de kullanacağız. Bu sebeple bu bilgiler önemlidir.

Oturum aç ?

Kullanıcı Adı:

Parola:

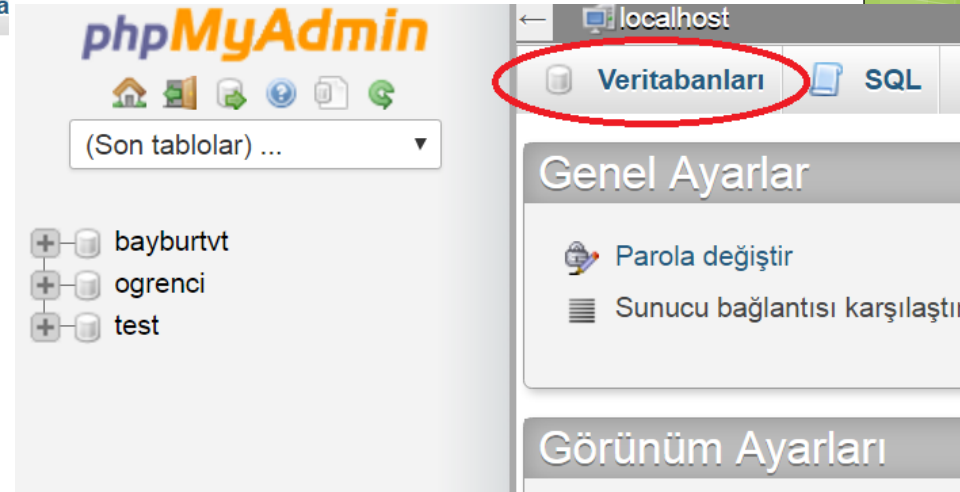
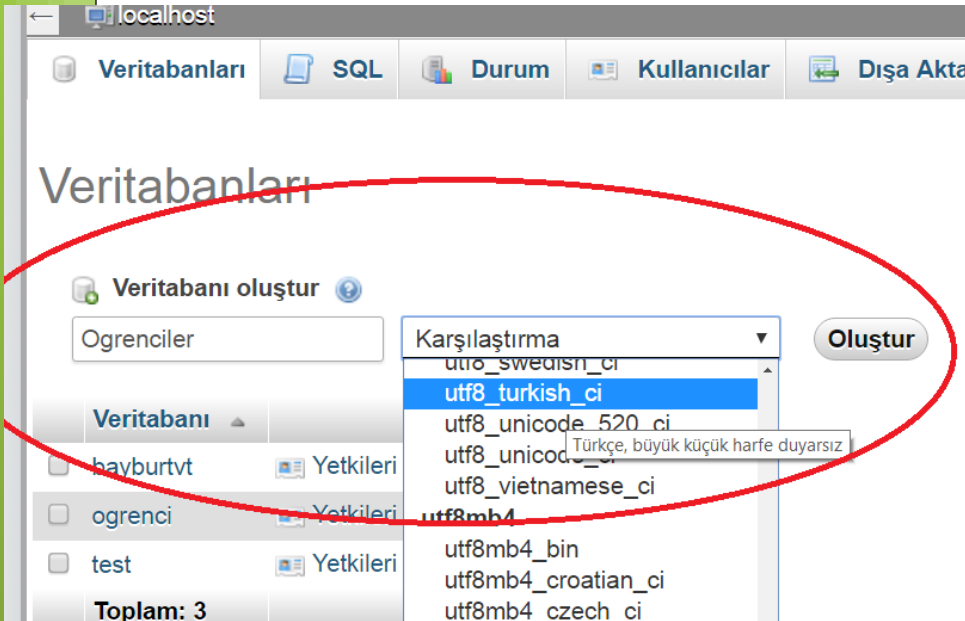
Default USBWebserver settings

Kullanıcı Adı:	root
Parola:	usbw
Mysql port	3307

Git

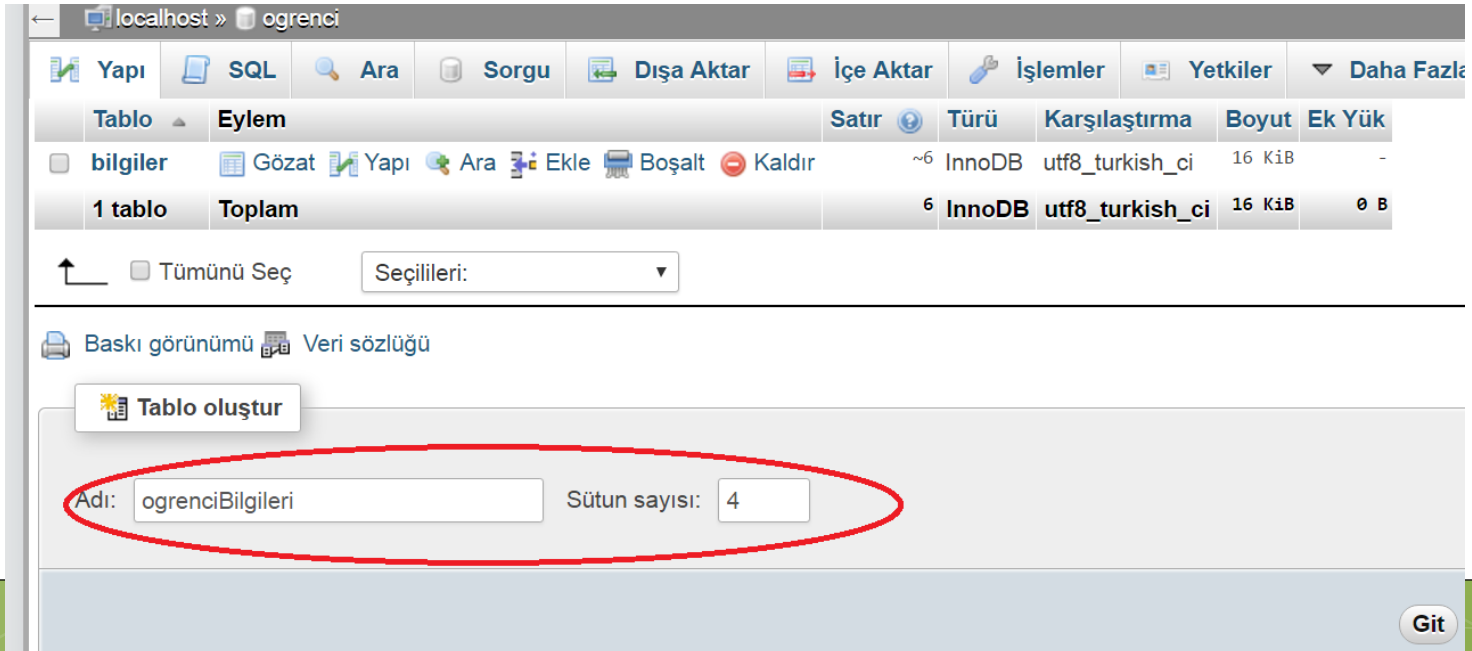
HTML, PHP ve MYSQL

Giriş yaptığımız panelin solunda var olan tüm veritabanı isimleri listelenir. Yeni veritabanı oluşturmak için üstteki veritabanları butonuna tıklanır ve karınıza gelen ekrandan veritabanı adını yazarak oluşturulur.



HTML, PHP ve MYSQL

Alttaki listeden oluşturulan veritabanının adına tıklayarak gelen ekrandan veritabanınıza tablo ekleyebilirsiniz. Tabloyu eklemeyen önce tabloda kaç adet alan olacağını yazmalısınız. Her bir veri için bir alan tanımlanır. Örneğin **ad, tc no, kullanıcı adı, şifre** alanları için 4 girmelisiniz



localhost » öğrenci

Yapı SQL Ara Sorgu Dışa Aktar İçe Aktar İşlemler Yetkiler Daha Fazla

Tablo	Eylem	Satır	Türü	Karşılaştırma	Boyut	Ek Yük
bilgiler	Gözet Yapı Ara Ekle Boşalt Kaldır	~6	InnoDB	utf8_turkish_ci	16 KiB	-
1 tablo	Toplam	6	InnoDB	utf8_turkish_ci	16 KiB	0 B

Tümünü Seç Seçimleri:

Baskı görünümü Veri sözlüğü

Tablo oluştur

Adı: öğrenciBilgileri Sütun sayısı: 4

Git

HTML, PHP ve MYSQL

Gelen ekranda tüm tanımlamak istediğiniz alanlar için bir değişken adı girip gerekli bilgileri doldurarak tablonuzun alanlarını oluşturabilirsiniz. Burada girdiğiniz isimlerin değişken kurallarına uygun olmasına dikkat etmelisiniz.

Gözet Yapı SQL Ara Ekle Dışa Aktar İçe Aktar İşlemler Tetikleyiciler

Tablo adı: 1 sütun ekle

Yapı

Adı	Türü	Uzunluk/Değerler	Varsayılan	Karşılaştırma	Öznelikler	Boş	İndeks	A_1	Yorumlar
adi	VARCHAR	100	Yok			<input type="checkbox"/>	---	<input type="checkbox"/>	
kullaniciAdi	VARCHAR	100	Yok			<input type="checkbox"/>	---	<input type="checkbox"/>	
sifre	VARCHAR	100	Yok			<input type="checkbox"/>	---	<input type="checkbox"/>	
tcNo	BIGINT	11	Yok			<input type="checkbox"/>	PRIMARY	<input type="checkbox"/>	

Tablo yorumları:

Depolama Motoru: Karşılaştırma:

PARTITION tanımı:

HTML, PHP ve MYSQL

Tablonuzu ilk defa oluşturduğunuzda. İçerisinde veri olmadığı için **Gözet** butonuna tıklasanız da bir şey göremezsiniz. Bunun için öncelikle veri ekleme işlemi yapmalısınız. **SQL** butonundan SQL sorgusu yazarak tablonuza veri ekleyebilirsiniz.

The screenshot shows a MySQL database management interface. The browser address bar indicates the local host and the database name 'ogrencibilgileri'. The interface has a menu bar with buttons for 'Gözet', 'Yapı', 'SQL', 'Ara', 'Ekle', 'Dışa Aktar', 'İçe Aktar', 'İşlemler', and 'Tetikleyiciler'. The 'SQL' button is circled in red. Below the menu bar, there is a text input field for the SQL query, which contains the following text: `INSERT INTO `ogrencibilgileri`(`adi`, `kullaniciAdi`, `sifre`, `tcNo`) VALUES ('Mutlu YAPICI','Happy','1234',12345678909)`. This text is also circled in red. Below the input field, there are buttons for 'SELECT *', 'SELECT', 'INSERT', 'UPDATE', 'DELETE', and 'Temizle'. The 'INSERT' button is circled in red. On the right side, there is a 'Sütun' (Columns) list with the following items: 'adi', 'kullaniciAdi', 'sifre', and 'tcNo'. At the bottom of the interface, there is a status bar with a checkbox for 'Bu sorguyu burada tekrar göster' and a 'Git' button, which is circled in red.

HTML, PHP ve MYSQL

Tablonuzu oluşturduğunuzda tablonuz listede görünecektir. Buradan seçerek tüm alanları ve varsa tüm verileri görebilirsiniz. Veritabanı oluşturma işlemi basit olarak bu şekilde gerçekleştirilmektedir.

Tablo	Eylem	Satır	Türü	Karşılaştırma	Boyut	Ek Yük
bilgiler	Gözet Yapı Ara Ekle Boşalt Kaldır	~6	InnoDB	utf8_turkish_ci	16 KiB	-
ogrencibilgileri	Gözet Yapı Ara Ekle Boşalt Kaldır	~0	InnoDB	utf8_turkish_ci	16 KiB	-
2 tablo	Toplam	6	InnoDB	utf8_turkish_ci	32 KiB	0 B

Tümünü Seç Seçilileri: ▼

Göster : Başlangıç satırı: 0 Satır sayısı: 30 Her 100 satırda bir başlıklar

+ Seçenekler

	adi	kullaniciAdi	sifre	tcNo
<input type="checkbox"/> Düzenle <input type="checkbox"/> Kopyala <input type="checkbox"/> Sil	Mutlu YAPICI	Happy	12345	12345678909

Tümünü Seç Seçilileri: Değiştir Sil Dışa Aktar

Göster : Başlangıç satırı: 0 Satır sayısı: 30 Her 100 satırda bir başlıklar

HTML, PHP ve MYSQL

PHP ile veritabanı sunucu bağlantısı için `mysql_connect()` fonksiyonunu kullanırız. Bağlantı gerçekleştirmek için gerekli verileri bu fonksiyona parametre olarak göndeririz. Sırası ile veritabanı sunucusunun adresini, veritabanı kullanıcı adını ve veritabanı kullanıcı şifresini girmemiz gerekmektedir.

```
mysql_connect('sunucu ip', 'kullanıcı adı', 'şifresi');
```

Veritabanı bağlantısı başarı ile gerçekleşirse fonksiyon sonucu olarak bağlantı verisi, gerçekleşmez ise **false** değeri dönmektedir.

HTML, PHP ve MYSQL

Aşağıdaki kod örneği bir veritabanı sunucu bağlantısı yapmaktadır. Eğer bağlantı gerçekleşmezse ekrana hata uyarısı çıkacaktır. Bağlantı sunucu adresi, kullanıcı adı veya şifresi hatalı olduğu için gerçekleşemeyebilir, bu verilere dikkat etmek gerekir.

```
$baglanti= mysql_connect("localhost","root","usbw");
```

```
if(!$baglanti)
```

```
{  
    echo"<h2>Veritabanı Bağlantı Hatası</h2>";  
}else{  
    echo"<h2>Veritabanına Bağlandı</h2>";  
}
```

HTML, PHP ve MYSQL

Veritabanı sunucu bağlantısı başarılı ise sırada veritabanını seçme işlemi vardır. Bir sunucuda birden fazla veritabanı bulunabilir bu yüzden hangi veritabanı ile çalışacaksa onu seçmemiz gerekir. Bunun için `mysql_select_db()` fonksiyonu kullanılır. Bu fonksiyonun ilk parametresi seçmek istediğiniz veritabanının adıdır. İkinci parametre ise daha önce yaptığımız sunucu bağlantısının değişkenidir. Sonuç olarak veritabanı başarı ile seçilirse **true** seçilemezse **false** değeri döner.

```
mysql_select_db("veritabanı adı", bağlantı değişkeni);
```

```
$vtsec= mysql_select_db("ogrenci",$baglanti);  
if(!$baglanti)  
{  
    echo "<h2>Veritabanı Seçme Hatası</h2>";  
}else{  
    echo "<h2>Veritabanına Seçildi</h2>";  
}
```

HTML, PHP ve MYSQL

Veritabanı başarı ile seçildikten sonra artık sorgu işlemlerine geçebiliriz. Seçtiğimiz veritabanındaki tablolar üzerinde sorgu yapmak için `mysql_query()` fonksiyonu kullanılır. Bu fonksiyonun parametresi yapılmak istenen sorgunun SQL cümlecığıdir. Bu cümlecik SELECT, UPDATE, INSERT, DELETE vb. olabilir. Sonuç olarak sorgu başarılı ise **true veya sorgu türüne göre veri**, döner başarısız ise **false** değeri döner. Bir sorgudan sonra işlemiz bittiğinde mutlaka bellekteki sorgu verilerini silmemiz gerekir yoksa bir müddet sonra şişme yapar. Bellekteki sorqu verilerini `mysql_free_result()` fonksiyonu ile silebiliriz.

```
$sql="SELECT * FROM 'ogrenciBilgileri'";
```

```
$sorgu= mysql_query($sql);
```

```
if($sorgu)
```

```
{
```

```
    echo"<h2>Veritabanı Sorgusu Başarılı</h2>";
```

```
    mysql_free_result($sorgu);
```

```
}else{
```

```
    echo"<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>";
```

```
}
```

HTML, PHP ve MYSQL

Yapılan sorgu sonucunda tabloda etkilenen satır sayısını `mysql_affected_rows()` fonksiyonu ile öğrenebilirsiniz. Eğer sorgunuz bu örnekte olduğu gibi bir `SELECT` sorgusuysa size sonuç olarak tablo satırları dönecektir. Kaç satır veri döndüğünü `mysql_num_rows()` fonksiyonu ile öğrenebilirsiniz.

```
$sql="SELECT * FROM 'ogrenciBilgileri'";  
$sorgu= mysql_query($sql);
```

```
if($sorgu)
```

```
{  
    echo "<h2>Veritabanı Sorgusu Başarılı</h2>";  
    echo "<h3>Sorgudan Etkilenen Satır Sayısı = ".mysql_affected_rows(). "</h3>";  
    echo "<h3>Sonuç Olarak Dönen Satır Sayısı = ".mysql_num_rows($sorgu). "</h3>";  
  
    mysql_free_result($sorgu);  
}else{  
  
    echo "<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>";  
}
```


HTML, PHP ve MYSQL

Başarılı bir şekilde sorgu yapmayı öğrendik şimdi de bu sorguların sonuçlarını nasıl işleyebiliriz onlara bakalım. Öncelikle SELECT sorgusundan gelen sonucumuzu ekranda göstereyim. Bunun için `mysql_fetch_rows()`, `mysql_fetch_assoc()` veya `mysql_fetch_array()` fonksiyonlarından birini kullanacağız. Aslında hepsi aynı fonksiyon ve hepsi sorgunun sonucunu bir diziye yükleyerek gönderiyor. Bizde dizilerle işlem yapıyormuş gibi veritabanı sonucunu işliyoruz. Tek farkları dizi indisleri (ofsetleri) dir. `mysql_fetch_rows()` fonksiyonu verileri diziye yüklerken indis olarak 0 dan başlayarak rakamları kullanıyor, `mysql_fetch_assoc()` fonksiyonu verileri diziye yüklerken indis olarak veritabanındaki alan adlarını kullanıyor, `mysql_fetch_array()` fonksiyonu ise çift indis kullanıyor yani ofset olarak hem 0 dan başlayarak rakamları hem de alan adlarını kullanıyor, bu sonucu fonksiyonda her veri iki defa gönderilmiş oluyor.

```
$sql="SELECT * FROM 'ogrenciBilgileri';  
$sorgu= mysql_query($sql);
```

```
if($sorgu)
```

```
{  
    echo"<h2>Veritabanı Sorgusu Başarılı</h2>";  
    echo"<h3>Sorgudan Etkilenen Satır Sayısı =".mysql_affected_rows(). "</h3>";  
    echo"<h3>Sonuç Olarak Dönen Satır Sayısı =".mysql_num_rows($sorgu). "</h3>";
```

```
    for($i=0;$i< mysql_num_rows($sorgu); $i++)
```

```
    {  
        $dizi= mysql_fetch_rows($sorgu);  
        echo "Adı = ".$dizi[0]."<br> Kullanıcı Adı : ".$dizi[1]. "<br> Kullanıcı Şifresi : ".$dizi[2]."<br> TC No : ".$dizi[3];  
    }
```

```
        mysql_free_result($sorgu);
```

```
    }else{
```

```
        echo"<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>";
```

```
    }
```

HTML, PHP ve MYSQL

Aynı örneği, **mysql_fetch_assoc()** fonksiyonuyla tekrar yazalım. Burada ofsetlere dikkat edin dizinin indisleri veritabanındaki alan adları olacak.

```
$sql="SELECT * FROM 'ogrenciBilgileri';  
$sorgu= mysql_query($sql);
```

```
if($sorgu)
```

```
{
```

```
echo"<h2>Veritabanı Sorgusu Başarılı</h2>";
```

```
echo"<h3>Sorgudan Etkilenen Satır Sayısı =".mysql_affected_rows(). "</h3>";
```

```
echo"<h3>Sonuç Olarak Dönen Satır Sayısı =".mysql_num_rows($sorgu). "</h3>";
```

```
for($i=0;$i< mysql_num_rows($sorgu); $i++)
```

```
{
```

```
    $dizi= mysql_fetch_assoc($sorgu);
```

```
    echo "Adı = ".$dizi['adi']."<br> Kullanıcı Adı : ".$dizi['kullaniciAdi']. "<br> Kullanıcı Şifresi :
```

```
 ".$dizi['sifre']."<br> TC No : ".$dizi['tcNo'];
```

```
    }
```

```
        mysql_free_result($sorgu);
```

```
}else{
```

```
    echo"<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>";
```

```
}
```

HTML, PHP ve MYSQL

Aynı örneği, `mysql_fetch_array()` fonksiyonuyla tekrar yazalım. Burada ofsetlere dikkat edin dizinin indisleri hem veritabanındaki alan adlarıyla hemde 0 dan başlayarak rakamlarla gelecek. Bu fonksiyonun ikinci parametresi `MYSQL_NUM`, `MYSQL_ASSOC`, `MYSQL_BOTH` değerleri alabilmektedir. Bu değerlere göre ofsetler sadece rakam, sadece alan adı yada her ikisi de olabilmektedir. Duruma göre istediğinizi kullanabilirsiniz.

```
$sql="SELECT * FROM 'ogrenciBilgileri";  
$sorgu= mysql_query($sql);
```

```
if($sorgu)
```

```
{
```

```
    echo"<h2>Veritabanı Sorgusu Başarılı</h2>";
```

```
    echo"<h3>Sorgudan Etkilenen Satır Sayısı = ".mysql_affected_rows(). "</h3>";
```

```
    echo"<h3>Sonuç Olarak Dönen Satır Sayısı = ".mysql_num_rows($sorgu). "</h3>";
```

```
    for($i=0;$i< mysql_num_rows($sorgu); $i++)
```

```
    {
```

```
        $dizi= mysql_fetch_array($sorgu, MYSQL_BOTH);
```

```
        print_r($dizi);
```

```
        echo "Adı = ".$dizi['adi']."<br> Kullanıcı Adı : ".$dizi['kullaniciAdi']."<br> Kullanıcı Şifresi : ".$dizi['sifre']."<br> TC
```

```
No : ".$dizi['tcNo'];
```

```
    }
```

```
        mysql_free_result($sorgu);
```

```
    }else{
```

```
        echo"<h2>Veritabanı Sorgusunda Hata Oluştü</h2>";
```

```
    }
```

HTML, PHP ve MYSQL

Şimdi bir veritabanına ekleme sorgusu yapalım. Açıkçası bundan sonra tek fark sadece SQL cümleciğinde olacaktır diğer kodlar aynı kalacak.

```
$sql="INSERT INTO `ogrenciBilgileri` (`adi`, `kullaniciAdi`, `sifre`, `tcNo`)
      VALUES('Ahmet', 'Ahmet0619', '12345','12345678999')";
$sorgu= mysql_query($sql);

if($sorgu)
{
    echo "<h2>Veritabanı Sorgusu Başarılı</h2>";
    echo "<h3>Sorgudan Etkilenen Satır Sayısı = ".mysql_affected_rows().
    "</h3>";
    echo "<h3>Sonuç Olarak Bir Veri Eklendi <h3>";
}
else{
    echo "<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>";
}
```

HTML, PHP ve MYSQL

Şimdi bir güncelleme (UPDATE) sorgusu yapalım. Açıkçası bundan sonra tek fark sadece SQL cümleciğinde olacaktır diğer kodlar aynı kalacak.

```
$sql="UPDATE `ogrenciBilgileri` SET `adi` ='Mehmet', `kullaniciAdi`=  
'MehmetSifresi' WHERE `tcNo`= 12345678999";  
$sorgu= mysql_query($sql);
```

```
if($sorgu)  
{  
    echo "<h2>Veritabanı Sorgusu Başarılı</h2>";  
    echo "<h3>Sorgudan Etkilenen Satır Sayısı =".mysql_affected_rows().  
    "</h3>";  
    echo "<h3>Sonuç Olarak".mysql_affected_rows(). " Veri Güncellendi <h3>";  
}  
else{  
    echo "<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>";  
}
```

HTML, PHP ve MYSQL

Şimdi de bir silme (DELETE) sorgusu yapalım. Açıkçası bundan sonra tek fark sadece SQL cümleciğinde olacaktır diğer kodlar aynı kalacak.

```
$sql="DELETE FROM `ogrenciBilgileri` WHERE `tcNo`=12345678999";  
$sorgu= mysql_query($sql);
```

```
if($sorgu)  
{  
    echo "<h2>Veritabanı Sorgusu Başarılı</h2>";  
    echo "<h3>Sorgudan Etkilenen Satır Sayısı =".mysql_affected_rows().  
    "</h3>";  
    echo "<h3>Sonuç Olarak".mysql_affected_rows(). " Veri Silindi <h3>";  
}  
else{  
    echo "<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>";  
}
```

HTML, PHP ve MYSQL

Şimdi de geçtiğimiz yılların en çok kullanılan ve en kolay hackleme yöntemlerinden olan SQL INJECTIONS dediğimiz SQL cümlecığı ekleme yöntemiyle bir örnek yapalım. Bunun için öncelikle kullanıcı girişi yapabildiğimiz bir form sayfası oluşturalım ve kullanıcı adımız ve şifremizle giriş yapalım. Görüldüğü gibi kullanıcı adı ve şifresi doğru ise Hoş geldin yanlıı ise yetkiniz yok yazısı çıkıyor.

Burada kullanıcı adı ve şifresi yerine SQL sorgusunu aldatmak için **1'OR'1'='1** verisi yazılırsa kullanıcı adı ve şifresini bilmeden yönetici girişi yapmış oluruz. Bu yöntemle veritabanına illegal olarak ulaşmış oluyoruz.

Veritabanına Bağlandı

Hoşgeldin Yönetici Mutlu YAPICI

Kullanıcı Adı :

Kullanıcı Şifresi :

Sorguyu gönder

Veritabanına Bağlandı

Giriş Yetkiniz Yok

Kullanıcı Adı :

Kullanıcı Şifresi :

Sorguyu gönder

HTML, PHP ve MYSQL

Kodlarımız:

```

<?php
$baglanti= mysql_connect("localhost","root","usbw");
if(!$baglanti)
{
    echo"<h2>Veritabanı Bağlantı Hatası</h2>";
}else{
    echo"<h2>Veritabanına Bağlandı</h2>";
    $vtsec= mysql_select_db("ogrenci",$baglanti);
    if(!$vtsec){ echo"<h2>Veritabanı Seçme Hatası</h2>"; }
    else{
        if(!empty($_POST['kadi']) && !empty($_POST['ksfr']) {
            $kadi= $_POST['kadi']; $ksfr=$_POST['ksfr'];
            $sql="SELECT * FROM `ogrencibilgileri` WHERE `kullaniciAdi`='".$kadi.'" AND `sifre`='".$ksfr.'"";
            $sorgu= mysql_query($sql);
            if($sorgu) {
                $veri= mysql_fetch_array($sorgu);
                if( mysql_num_rows($sorgu)>0)

                    echo"<h2>Hoşgeldin Yönetici ".$veri['adi']."</h2>";
                else
                    echo"<h2>Giriş Yetkiniz Yok</h2>";
                mysql_free_result($sorgu);
            }else{ echo"<h2>Veritabanı Sorgusunda Hata Oluşturdu</h2>"; }
        }
    }
    mysql_close($baglanti);
}
?>
<form action="vtbaglan.php" method="POST">
    <table width="400px" height="150px">
        <tr> <td> Kullanıcı Adı : </td><td> <input type="text" name="kadi"></td></tr>
        <tr> <td> Kullanıcı Şifresi : </td><td> <input type="password" name="ksfr"></td></tr>
        <tr> <td> </td><td> <input type="submit" name="btn"></td></tr>
    </table>
</form>

```


HTML, PHP ve MYSQL

Bu tür kodlardan korunabilmemiz için php de çeşitli fonksiyonlar vardır, bu fonksiyonlar veritabanına zararlı kod içeren verilerin gönderilmesini engeller bunlardan biri **mysql_real_escape_string()** fonksiyonudur. Formdan gelen verileri bu fonksiyondan geçirerek alırsanız SQL INJECTION lardan krunursunuz.

```
$kadi=mysql_real_escape_string($_POST['kadi']);  
$ksfr=mysql_real_escape_string($_POST['ksfr']);
```

HTML, PHP ve MYSQL





Şimdi de veritabanına veri eklerken ve yapısını oluştururken dikkat edilmesi gerekenlerden bahsedeceğim. Veritabanında ve sayfalarda sıklıkla karşılaşılan sorunlardan biri de Türkçe karakter sorunudur. Eğer başlangıçta hem veritabanında hem de sayfalarda karakter kodlamalarını ayarlamazsak sıklıkla bu sorunla karşılaşırız. Şimdi birkaç örnekle bu soruna bakalım.

Veritabanımızı utf-8 e göre ayarladık. Kodlama türü ANSI olan bir sayfadan veri ekleme yapalım ve veritabanındaki yapıya bakalım.

HTML, PHP ve MYSQL

Görüldüğü gibi veritabanı utf-8 ancak ANSI olan bir sayfadan veriler gönderilince veritabanında Türkçe karakterler düzgün görünmemektedir.

+ Seçenekler

				adi	kullaniciAdi	sifre	tcNo
<input type="checkbox"/>	 Düzenle	 Kopyala	 Sil	Mutlu YAPICI	Happy	12345	12345678909
<input type="checkbox"/>	 Düzenle	 Kopyala	 Sil	Pener Göktürk	Söđüt	ApýkÝpek	12345678994
<input type="checkbox"/>	 Düzenle	 Kopyala	 Sil	Mehmet	Mehmet0619	12345	12355678909

Bu sorunun üstesinden gelebilmek için MYSQL sorgusunu gönderdiğimiz sayfada **mysql_query()** den hemen önce karakter kümesini aşağıdaki kodlarla tanımlamalıyız.

```
mysql_query("SET NAMES 'utf8'");
```

```
mysql_query("SET CHARACTER SET utf8");
```

```
mysql_query("SET COLLATION_CONNECTION = 'utf8_turkish_ci'");
```

HTML, PHP ve MYSQL













Önceki sunumda bulunan karakter kümelerini tanımlasanız bile yine veritabanına Türkçe karakterlerin eklenmediğini göreceksiniz bunun sebebi ise karakter kümesini tanımladığınız sayfanın karakter kodunun ANSI olmasıdır. O sayfanın da karakter kodunu utf8 yapmalıyız. Bunun için ister header() fonksiyonunu ister meta tagleri isterseniz de notepad++ ı kullanabilirsiniz.

```
<meta charset="utf-8"/>
```

```
header('Content-type: text/html; charset=utf-8');
```

Mutlu YAPICI	Happy	12345	12345678909
Äener GÄ¼ktÄ¼rk	SÄ¼Ä¼Ä¼t	AÄ¼Ä¼kÄ¼pek	12345678994
Mehmet	Mehmet0619	12345	12355678909
	S	A	13345678994

+ Seçenekler

	adi	kullaniciAdi	sifre	tcNo
<input type="checkbox"/>   	Mutlu YAPICI	Happy	12345	12345678909
<input type="checkbox"/>   	Pener Göktürk	Södüüt	ApykYpek	12345678994
<input type="checkbox"/>   	Mehmet	Mehmet0619	12345	12355678909
<input type="checkbox"/>   		S	A	13345678994

HTML, PHP ve MYSQL

Sonuç olarak tüm ayarları yaptıysanız hem veritabanında hem de internet sayfanızda Türkçe karakter sorunu çözülmüş olacaktır. Son eklenen verinin iki yerde de karakterleri düzgün ancak önceden eklediklerimizin düzelmesi için silip yeniden eklenmeleri gerekir.

İnternet Sayfası

Veritabanı

Mutlu YAPICI	Happy	12345	12345678909
Pener Göktürk	Söđüt	AşıkYpek	12345678994
Mehmet	Mehmet0619	12345	12355678909
	S	A	13345678994
Sengül Göktürk	Söğütözüm	Aşıkİpek	14445678994

+ Seçenekler				adi	kullaniciAdi	sifre	tcNo
<input type="checkbox"/>				Mutlu YAPICI	Happy	12345	12345678909
Daha önce Eklenen Veri				Pener Göktürk	Söđüt	AşıkYpek	12345678994
<input type="checkbox"/>				Mehmet	Mehmet0619	12345	12355678909
<input type="checkbox"/>					S	A	13345678994
<input type="checkbox"/>				Şengül Göktürk	Söğütözüm	Aşıkİpek	14445678994
				Son Eklenen Veri			
<input type="checkbox"/>	Tümünü Seç			Seçilleri:			

KAYNAKLAR

- İnternet ortamı
- PHP ve AJAX Haydar TUNA
- A'dan Z'ye PHP Rıza ÇELİK

