**PHP’de MySQL veritabanı işlemleri (SELECT, INSERT, UPDATE, DELETE)**

**CRUD – CREATE – READ – UPDATE- DELETE**

PHP’de MySQL veritabanı işlemlerine kaldığımız yerden devam ediyoruz. Bugün MySQL veritabanından temel SQL komutlarını kullanarak (SELECT, INSERT, UPDATE, DELETE) basit bir örnek yapacağız.

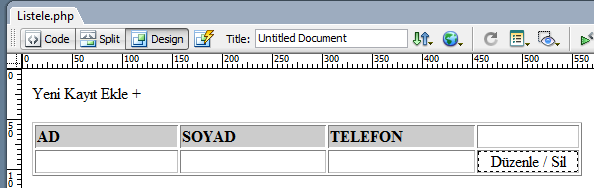
Uygulamamız 4 bölümden oluşacaktır.

1. Kayıt listeleme,
2. Kayıt ekleme,
3. Kayıt düzenleme,
4. Kayıt silme

Veritabanımız, önceki MySQL örneklerimizde kullandığımızın aynısı fakat ek olarak VARCHAR tipinde 15 karakterlik “Telefon” alanı oluşturun.

**1. Kayıt Listeleme**

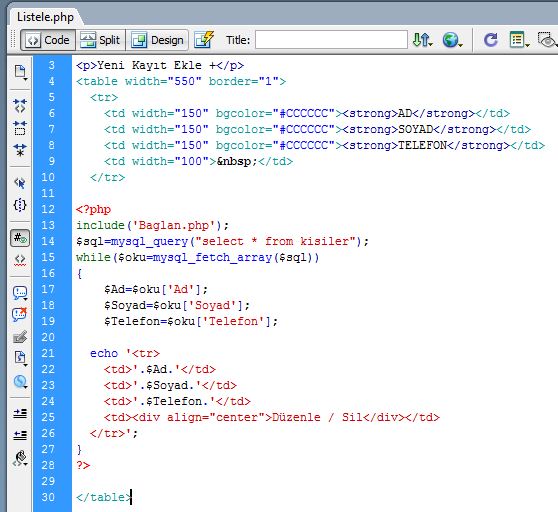
Öncelikle kayıtlarımızın listeleneceği sayfayı “Listele.php” adında oluşturalım. MySQL veritabanından veri okuma işleminden daha önceden [bahsetmiştik](http://blog.hakanhitay.net/php/phpde-mysql-veritabani-islemleri-select-insert-update-delete/2011/07/23/php%e2%80%99de-mysql-veritabanindan-veri-okuma/). Şimdi okuduğumuz verileri bir tablonun içerisine düzenli bir şekilde listeleteceğiz. Bunun için 2 satır 3 sütunluk bir tablo oluşturalım.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-1.png)

Gelelim PHP kısmına..

Veritabanından verilerimizi okuyalım ve tablomuzun 2. satırındaki hücrelere yazdıralım. Tablonun 1. Satırında sütun isimleri yazacak olduğu için verileri okurken kullandığımız döngünün dışında tutacağız. Veriler sadece döngü yardımı ile 2. satırdan itibaren yazılacak.

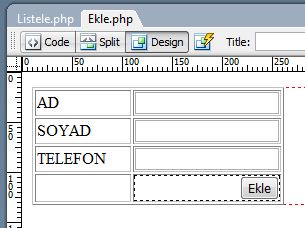
[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-2.jpg)

Yazdığımız kodu kontrol edelim. Ekran görüntüsü aşağıda.[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-3.jpg)

Uygulamamızın ilk bölümünü böylece bitirmiş olduk.

**2. Kayıt Ekleme**

2. bölümde de, 1. bölümde oluşturduğumuz “Yeni Kayıt Ekle +” yazısına link vererek yeni bir sayfa “Ekle.php” adında bir sayfa görüntületiyoruz. Bu sayfa için bir HTML form oluşturalım. Formu aşağıdaki gibi yapıyoruz. Formun action bölümünü “Ekle2.php” yapalım. Oluşturacağımız formu sonraki bölümlerimizde de kullanacağız.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-4.jpg)

1. bölümde oluşturduğumuz sayfada yer alan “Yeni Kayıt Ekle +” yazısına link vererek bu sayfaya erişimi sağlayalım.

Bu bölümün de PHP kodunuz yazmaya başlayalım. Formdaki verileri “Ekle2.php” ye göndereceğimizi belirtmiştik. “Ekle2.php” adında bir PHP dosyası açalım.

İlk olarak gönderdiğimiz verileri alalım.

|  |  |
| --- | --- |
| 1  2  3 | $Ad=$\_POST["Ad"];  $Soyad=$\_POST["Soyad"];  $Telefon=$\_POST["Telefon"]; |

Verilerimizi aldıktan sonra sıra geldi bir SQL komutu olan “insert” i kullanmaya. “insert” bildiğiniz üzere veritabanımıza yeni kayıt eklememize yarar. Biz de aldığımız verileri bu komut ile veritabanımıza ekleyeceğiz. PHP’de SQL komutlarını *mysql\_query()* fonksiyonunu kullanıyorduk. SQL ifademizi yazıyoruz.

|  |  |
| --- | --- |
| 1 | $Komut=mysql\_query("insert into kisiler values('','$Ad','$Soyad','$Telefon')"); |

İlk alanımız otomatik sayı özelliğinde olduğu için SQL komutumuzu yazarken o bölümü boş bıraktık. Formdan aldığımız verileri atadığımız değişkenlerimizi values() kısmında yerleştirdik.

SQL komutumuzdan dönecek değeri (True yada False) $Komut değişkenine atadık. Kaydımızın sorunsuz bir şekilde gerçekleşmesi halinde $Komut değişkenimizin değeri “True” olacağından if() kullanarak ekrana “Kaydınız başarıyla gerçekleşmiştir.” yazdırıyoruz.

Ardından mevut sayfamızı (Ekle2.php), 1. bölümde hazırlamış olduğumuz “Listele.php” sayfasına yönlendiriyoruz. PHP’de sayfa yönlendirmeyi 2 şekilde yapabiliriz.

**1. Header(“refresh”) yöntemi ile;**

Bu yöntem ile yönlendirilecek olan sayfaya belirlenen süre sonunda yönlendirebiliriz. Kullanımı aşağıdaki gibidir.

|  |  |
| --- | --- |
| 1 | header("refresh:2; url=url adı yada sayfa adı"); |

Buradaki “2” değeri, kodun işlenmeye başladıktan 2 saniye sonra belirtilen sayfaya yönlendirileceği anlamındadır.

**2. Header(‘Location’) yöntemi ile;**

Bu yöntemde yönlendirilecek olan sayfaya zaman belirtmeksizin direkt olarak yönlendirme işlemi yapılır. Bu yöntemin kullanımı da aşağıdaki gibidir.

|  |  |
| --- | --- |
| 1 | Header(“Location:url adı yada sayfa adı”); |

Header fonksiyonunu kullandığınız sayfada “**Warning**: Cannot modify header information – headers already sent by…” şeklinde bir hata mesajı alabilirsiniz. Hatayı gidermek istediğiniz sayfanın başına,

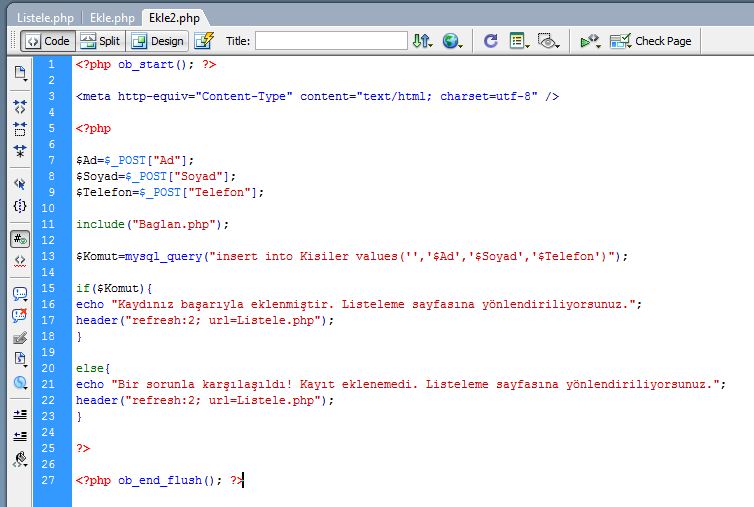
|  |  |
| --- | --- |
| 1 | <?php ob\_start(); ?> |

sonuna ise,

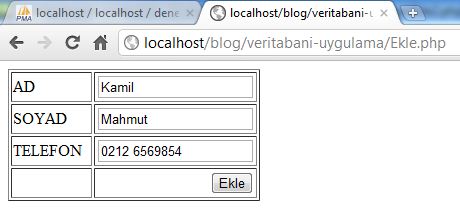
|  |  |
| --- | --- |
| 1 | <?php ob\_end\_flush(); ?> |

ekleyin.

“Ekle2.php” sayfamızın kodlama kısmı bitmiş bulunmakta. Kodlar aşağıdaki gibidir.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-5.jpg)

Yazdığımız kodu deneme zamanı.. Öncelikle “Ekle.php” sayfasını açıyoruz.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-6.jpg)

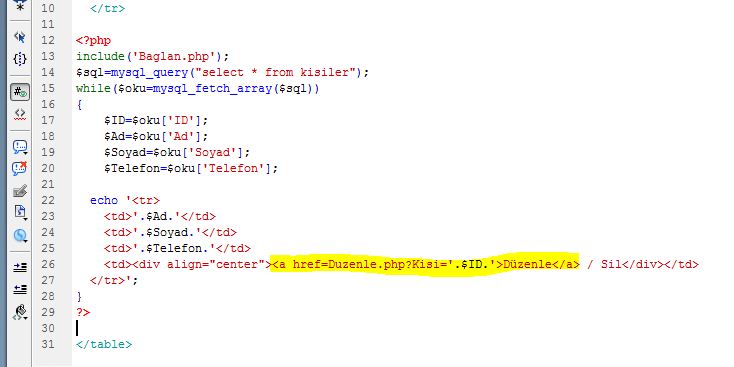
Ve yazdığımız kod sorunsuzca çalışıyor. :-)

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-7.jpg)

**3. Kayıt Düzenleme**

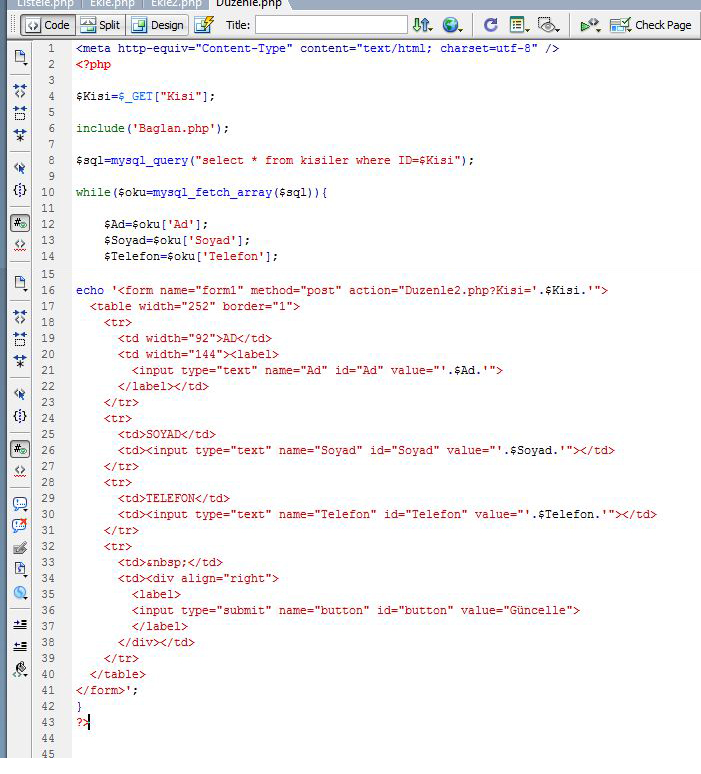
Kayıt düzenleme için öncelikle 1. bölümde oluşturduğumuz “Listele.php” sayfasında kayıtlarımızın yanındaki “Düzenle” yazısına link verelim. (Duzenle.php)

Veritabanımızda birincil anahtar olarak “ID” alanını seçmiştik. “ID” alanından dönen değerleri “Düzenle” linkinin sonuna “?” operatörünü kullanarak ekleyelim.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-8.jpg)

“Duzenle.php” sayfamıza geçelim. Bu sayfa için bir form oluşturacağız. Tekrardan form oluşturmamak için 2. bölümde oluşturduğumuz “Ekle.php” den alıyoruz. Fakat formun action bölümünü “Duzenle2.php” yapmayı unutmayalım. Ayrıca “Duzenle2.php” sayfasına gönderecek olduğumuz ID bilgisini de “?” operatörünü kullanarak formun action kısmında belirtelim.

Gönderdiğimiz ID değerine göre veritabanında eşleşen kişinin bilgilerini text kutularının “value” kısmına yazdırıyoruz.

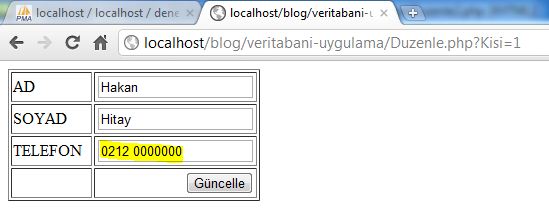
[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-9.jpg)

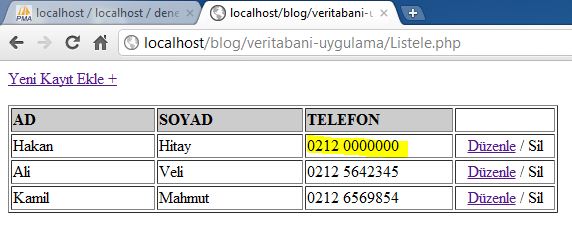
Gelelim “Duzenle2.php” ye..

Öncelikle “Duzenle.php” den gönderdiğimiz güncel verileri çekiyoruz. Sonrasında SQL komutumuzu yazıyoruz. Sorgumuzun sonunda ”where” koşulunu belirtmeyi de unutmuyoruz. Aksi takdirde güncelleme işlemi tüm kayıtlarımızda gerçekleşir. :-)

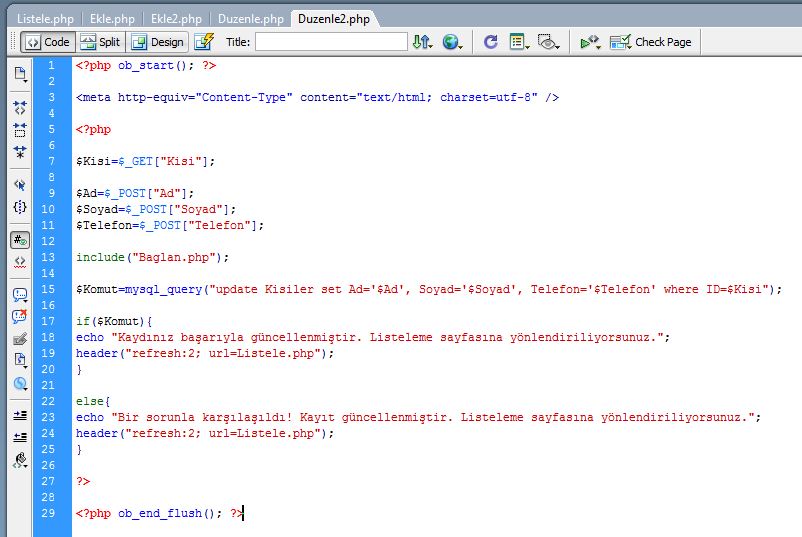
|  |  |
| --- | --- |
| 1 | $Komut=mysql\_query("update Kisiler set Ad='$Ad', Soyad='$Soyad', Telefon='$Telefon' where ID=$Kisi"); |

Yazdığımız kodu deneyelim..

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-11.jpg)

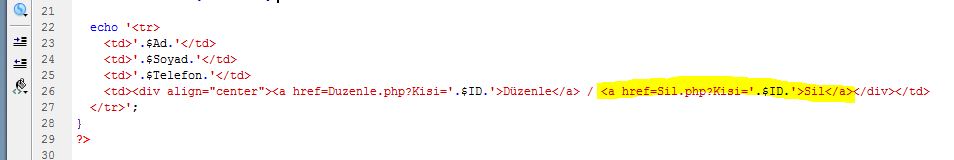
[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-12.jpg)

Yazdığımız kod başarılı bir şekilde çalışıyor. Genel olarak “Duzenle2.php” nin kodları aşağıdaki gibidir.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-10.jpg)

**4. Kayıt Silme**

Ve sıra bugünkü uygulamamızın son kısmına geldi. İlk olarak düzenleme bölümünde yaptığımız gibi “Listele.php” de “Sil” yazısına “Sil.php” ye gidilecek şekilde link verelim. Linkimizin sonuna, “?” operatörünü kullanarak kişinin ID bilgisini de eklemeyi unutmayalım.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-13.jpg)

“Sil.php” sayfamızı oluşturmaya başlayalım.

İlk olarak “Listele.php” sayfasındaki linkten gönderdiğimiz kişinin ID bilgisini GET yöntemiyle çekelim.

Ardından SQL ifademizi yazalım.

|  |  |
| --- | --- |
| 1 | $Komut=mysql\_query("delete from Kisiler where ID=$Kisi"); |

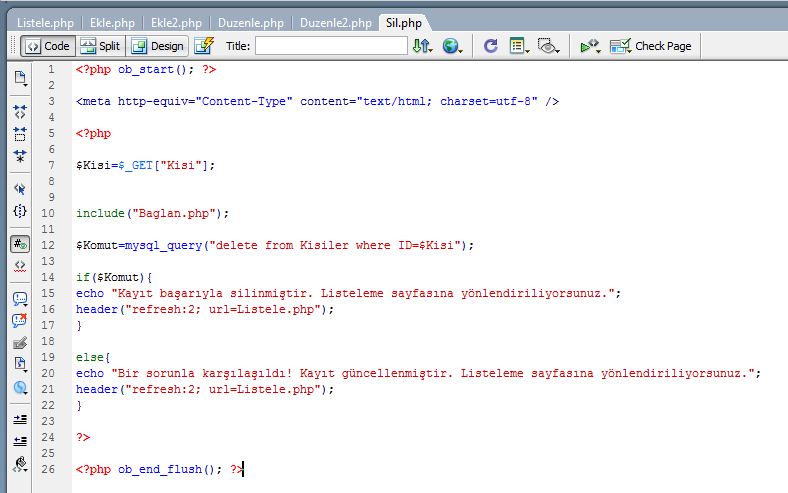
Burada da tekrar belirtmek istiyorum. SQL ifademizde “where” koşulunu belirtmeyi kesinlikle unutmayalım. Koşul belirtmezsek SQL sorgumuz tüm kayıtlar için geçerli olur. Biz şu anda silme işlemi yaptığımız için bunun sonucunda tüm kayıtlarımız veritabanından silinir.

Silme işleminin başarıyla sonuçlanması durumunda ekrana “Kayıt başarıyla silinmiştir. Listeleme sayfasına yönlendiriliyorsunuz.” yazdırdık ve “Listele.php” sayfamıza yönlendirme yaptık.

Yazdığımız kodu deneyelim.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-15.jpg)[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-16.jpg)

Silme kodumuz da başarılı bir şekilde çalışıyor. “Sil.php” kodları da aşağıdaki gibidir.

[](http://blog.hakanhitay.net/wp-content/uploads/2011/07/veritabani-uygulama-14.jpg)

Böylece PHP’de MySQL veritabanı işlemlerini bitirmiş olduk.