WEB TABANLI PROGRAMLAMA

BÖLÜM 14 PHP İLE MYSQL KULLANIMI MYSQL BAĞLANTILI PHP UYGULAMALARI GELİŞTİRME

Prof. Dr. Turgay Tugay Bilgin turgay.bilgin@btu.edu.tr



GENEL BAKIŞ...

14.1) Veri Tabanı Bağlantılı Tasarım: 3 – Katmanlı Mimari

14.2) MySQL Veri Tabanı Sunucusu

14.3) phpMyAdmin Kullanarak MySQL'e Bağlanmak

14.4) MySQL Sunucuda Veri Tabanı ve Tablo Oluşturma

14.5) SQL Dili Kullanarak Veri Tabanı ve Tablo Oluşturmak

14.6) PHP ile MySQL Veri Tabanına Bağlanmak

14.7) PHP ile MySQL Veri Tabanına Bilgi Girmek

14.8) PHP ile MySQL Veri Tabanındaki Tablolardan Bilgi Okumak

14.9) PHP ile MySQL Veri Tabanındaki Tablolardan Bilgi Silmek

14.10) PHP ile MySQL Veri Tabanındaki Tablolarda Bilgi Güncelleme

14.11) Özet

PHP dilinde bilgileri sunucuda kalıcı depolamak için önceki ünitelerde metin dosyaya yazma ve okuma işlemleri gösterilmişti. Bu bölümde ise metin dosyalar yerine veri tabanı sunucusuna bilgi yazma gösterilecektir. Veri tabanı sunucularına bilgi yazabilmek için veri tabanı sorgulama dili olan SQL (*Structured Query Language – Yapısal Sorgulama Dili*) dilinin temel bazı komutlarının bilinmesi konunun daha kolay anlaşılmasını sağlayacaktır. Bu ünitede, web tabanlı MySQL yönetim programı olan phpMyAdmin ile SQL bilmeyenlerin de anlayabileceği şekilde veri tabanı ve tablo oluşturma işlemleri açıklanmıştır. Bu ünitedeki örneklerde veri tabanında tablo oluşturma, bilgi ekleme, silme gibi temel işlemlere ilişkin SQL deyimleri kısaca bahsedilmiş ve kod örnekleri verilmiştir.

(14.1) VERİ TABANI BAĞLANTILI TASARIM: 3 KATMANLI MİMARİ

 Şimdiye kadar gördüğümüz tüm web uygulamaları temel olarak 2 katmanlı yapıdadır. Bunlardan ilk web tarayıcıda çalışan kodlar, ikincisi ise web sunucuda çalışan kodlardır. Bu tür mimarilere 2 katmanlı mimari denir. Şekil 14.1'de 2 katmanlı mimari diyagramı görülmektedir.



Şekil 14.1 İki katmanlı mimari diyagramı

 Veri tabanı bağlantılı sistemler ise 3 katmanlı mimaridedir. 2 katmanlı mimariye ek olarak veri tabanında çalışan kodlar da sisteme dâhil olmaktadır.



- Şekil 14.2 Üç katmanlı mimari diyagramı
- Bu yapıda web sunucu yazılımı PHP yorumlayıcıya, PHP yorumlayıcı ise veri tabanı sunucuya bağlantı kurmaktadır.
 Şekil 14.2'de 3 katmanlı mimari diyagramı görülmektedir.
 Bu ünitede PHP yorumlayıcı ile veri tabanı sunucusu arasındaki bağlantı gerçekleştirme anlatılacaktır.

(14.2) MYSQL VERİ TABANI SUNUCUSU

- MySQL bir veri tabanı sunucusu olup PHP ile birlikte en fazla kullanılan veri tabanı olma özelliğine sahiptir. PHP ile Microsoft veya Oracle gibi başka üreticilerin veri tabanlarına bağlanmak da mümkündür. Fakat, MySQL çok daha yaygın olarak tercih edilmektedir. Bunun en önemli sebebi, PHP yorumlayıcısının içinde MySQL veri tabanına bağlanmak için gerekli eklentiler hazır olarak gelmesidir. Bu sayede herhangi bir ek programa ihtiyaç kalmadan MySQL bağlantıları gerçekleştirilebilmektedir.
- MySQL veri tabanı sunucusu Windows, Linux ve MacOS işletim sistemlerinde çalışabilir. MySQL veri tabanı sunucusunun standart bir görsel arayüzü yoktur. Normal kurulumda görsel arayüz kurulu olarak gelmez. Dolayısıyla MySQL çoğunlukla komut satırı yönetim paneli aracılığı ile kullanılır (Şekil 14.3). WampSERVER yazılımında MySQL komut satırını açmak için WampSERVER ikonu sol tıklanır, açılan menüden MySQL -> MySQL Console yolu izlenebilir. Fakat bu şekilde kullanmak zor olduğundan MySQL için farklı firmalar tarafından üretilmiş birçok görsel arayüz bulunmaktadır.



Şekil 14.3. MySQL sunucusu komut satırı arabirimi.

• Bunlardan bazıları MySQL Workbench, HeidiSQL, Navicat olarak verilebilir. Bunların dışında web tabanlı yönetim panelleri de vardır. Bunlardan en bilineni phpMyAdmin olarak verilebilir. Derslerde kullandığımız WampSERVER paket yazılımında phpMyAdmin geldiği için uygulamalarımızı phpMyAdmin ile gerçekleştireceğiz

(14.3) PHPMYADMIN KULLANARAK MYSQL'E BAĞLANMAK

phpMyAdmin aracının kendisi de PHP ile geliştirilmiştir. MySQL veri tabanının yönetimi için görsel bir arayüz sunar. Web arayüzü aracılığıyla yapılan işlemler arka planda SQL deyimleri haline dönüştürülerek MySQL sunucuya iletilir. phpMyAdmin programını başlatmak için Şekil 14.4'de görüldüğü gibi WampSERVER ikonunu tıklayarak açılan menüden phpMyAdmin seçilir. Bu durumda varsayılan web tarayıcınız http://localhost/phpmyadmin/ adresini açarak phpMyAdmin programını ekrana getirir.

• phpMyAdmin programı aracılığıyla MySQL veri tabanı üzerinde veri tabanı oluşturma ve silme, tablo oluşturma ve silme, tabloya alan ekleme ve çıkarma, SQL sorguları çalıştırma gibi her türlü veri tabanı işlemi gerçekleştirilebilmektedir.



(14.4) MYSQL SUNUCUDA VERİ TABANI VE TABLO OLUŞTURMAK

- PHP ile MySQL sunucuya bağlanabilmek için MySQL sunucuda bir veri tabanı ve içinde en az bir tablo bulunmalıdır. Bundan sonraki örneklerimizde kullanabilmek için bir veri tabanı ve bir tabloyu phpMyAdmin programı ile oluşturacağız. Öncelikle Şekil 14.4'teki gibi phpMyAdmin programını başlatınız.
- Örnek vertibanı olarak bir firmadaki personel kayıtlarını tutacağız. Bunun için öncelikle:
 - "Personel" adlı bir veri tabanı,
 - Veri tabanı içine "memurlar" adlı bir tablo,
 - Tabloda aşağıdaki alanlar oluşturulacaktır.

✓ "soyad",

Örnek 14.1. phpMyAdmin ile veritabanı ve tablo oluşturma

- Şimdi aşağıdaki adımları phpMyAdmin üzerinde gerçekleştiriniz:
- 1) Şekil 14.5'te dikdörtgen şeklinde işaretlenmiş olan "Yeni" bölümünü tıklayınız.



Şekil 14.5. phpMyAdmin programında veri tabanı oluşturmada ilk adım

Şekil 14.6'da 1 numaralı dikdörtgen ile işaretlenmiş bölüme veri tabanı adını "personel" olarak yazınız. Veri tabanında tutulacak bilgilerde "ĞÜŞİÖÇğüşiöç" gibi Türkçe karakterlerde problem oluşmaması için "Karşılaştırma" bölümüne "utf8_turkish_ci" değerini seçerek "Oluştur" butonunu tıklayınız.



Şekil 14.6. phpMyAdmin programında veri tabanı oluşturmada ikinci adım

3) Üçüncü adımda "personel" veri tabanı içinde "memurlar" tablosu oluşturulacaktır. Şekil 14.7'de 1 numarayla işaretlenmiş olan bölüme tıklayarak veri tabanı adını "personel" olarak seçiniz. Sonra, 2 nolu alana "memurlar" yazınız ve 3 nolu alana 5 değerini giriniz. Tabloya 5 adet alan ekleneceği için 5 değeri giriliyor.
4 nolu bölümde gösterilen "Git" butonuna tıklayınız.



Şekil 14.7. phpMyAdmin programında veri tabanı oluşturmada üçüncü adım

4) Bu adımda "memurlar" tablosu içinde alanlar oluşturulacaktır. Şekil 14.8'de gri işaretlenmiş olan "Adı" sütunundaki metin kutularına oluşturulacak tablodaki alanların adlarını, "Türü" sütunundaki metin kutularına veri türlerini ve uzunluklarını şekildeki gibi giriniz

Dikkat George Tablolardaki alanların veri türleri

Veritabanı tablolarında her bir alanın içinde tutulacak verinin tipi sunucu tarafından bilinmelidir. Veri tipler genellikle INT veya VARCHAR değerleri alırlar. INT değeri tamsayı içeren değerleri ifade ederken VARCHAR ise değişken uzunlukta harf veya karakterlerden oluşan verileri ifade eder. Şekil 14.8'de INT değer içeren alanların uzunluğu 4 birim, VARCHAR değer içeren alanların uzunluğu 25 birim olarak oluşturulmuştur. Bu örnekte INT alanlara 4 ve VARCHAR alanlara 25'den daha kısa bilgi girebilirsiniz fakat daha uzun bilgi giremezsiniz.

• Tablolarda en az 1 alan birincil anahtar olmalıdır. Birincil anahtar, tablodaki tüm değerlerin birbirinden farklı olmasını sağlayan sayısal değerdir. Bu tabloda "**memur_id**" alanı birincil anahtar (primary key) olarak belirlenmiştir. Böylece, aynı "**memur_id**" değerine sahip iki kayıt girilmeye çalışılırsa veri tabanı buna izin vermeyecektir. "**memur_id**" alanını birincil anahtar yapmak için "Index" alanında "PRIMARY" seçiniz ve otomatik artan sayılar olarak değer alması için **A_I** (auto increment) kolonundaki onay kutusunu seçiniz ve altta sağdaki "Kaydet" butonunu tıklayınız.

			Y	арі 😡					
Adı	Türü 😣	Uzunluk/Değerler 😣	Varsayılan 😡	Karşılaştırma	Öznitelikler	Boş	Index	A_I	Yorumlar
memur_id	INT	4	Yok				PRIMARY	V	
ad	VARCHAR	25	Yok						
soyad	VARCHAR	25	Yok 💌						
birim	VARCHAR	20	Yok 💌		V		💌		
maas	INT	5	Yok						

Şekil 14.8. phpMyAdmin programında veri tabanı oluşturmada dördüncü adım

• Bu adımlar tamamlandığında personel veri tabanı içinde 5 adet alana sahip memurlar tablonuz oluşacaktır.

(14.5) SQL DİLİ KULLANARAK VERİ TABANI VE TABLO OLUŞTURMAK

SQL dili komutları kullanarak da MySQL sunucuda veri tabanı ve tablo oluşturulabilir. Bölüm 14.4'de phpMyAdmin üzerinde gerçekleştirilen işlemlere ait SQL kodu aşağıda verilmiştir.

```
CREATE DATABASE `personel` DEFAULT CHARACTER SET utf8
COLLATE utf8_turkish_ci;
USE `personel`;
CREATE TABLE `memurlar` (
  `memur_id` int(4) NOT NULL AUTO_INCREMENT,
  `ad` varchar(25) NOT NULL,
  `soyad` varchar(25) NOT NULL,
  `birim` varchar(25) NOT NULL,
  `maas` int(5) NOT NULL,
  PRIMARY KEY (`memur_id`)
);
```

Kod 14.1. SQL dili ile personel veritabanı ve memurlar tablosu oluşturulması



Şekil 14.9. phpMyAdmin ile SQL dili komutları kullanımı

Örnekteki kodları Şekil 14.3'de görülen MySQL konsoluna giriş yaparak buraya yazar ve "Enter" tuşuna basarsanız veri tabanı ve tabloları oluşturabilirsiniz. Bunun haricinde phpMyAdmin ekranında Şekil 14.9'da görüldüğü gibi *sol menüde hiçbir veri tabanı seçili değilken* SQL sekmesini tıklayarak ekrana gelen metin kutusuna yazabilir ve en alt sağ köşedeki "git" butonunu tıklayarak çalıştırabilirsiniz. 14

(14.6) PHP İLE MYSQL VERİ TABANINA BAĞLANMAK

• PHP ile MySQL veri tabanına bağlanmak için PHP içinde hazır olarak mysqli_connect() fonksiyonu bulunur. Bu fonksiyonun kullanımı aşağıdaki gibidir:

mysqli_connect(server,user,password,database);

Tablo 14.1 mysqli_connect parametreleri ve tanımları

Parametre	Тапіті
server	Buraya MySQL sunucunun adı veya IP adresi yazılır. Eğer PHP ile
Derver	MySQL aynı makinada ise server adı localhost'dur.
	Buraya, MySQL sunucuya hangi kullanıcı olarak bağlanılacak ise
user	onun adı yazılır. Bu kullanıcı MySQL sunucuda var olan bir kullanıcı
	olmalıdır. Wampserver'da varsayılan MySQL kullanıcısı root 'dur.
	Buraya, MySQL sunucuya hangi kullanıcı olarak bağlanılacak ise
password	onun şifresi yazılır. Bu kullanıcı mysql sunucuda var olan bir
	kullanıcı olmalıdır. Wampserver'da varsayılan MySQL kullanıcısı
	root'un şifresi yoktur (şifresizdir).
database	Buraya, MySQL sunucuda hangi veri tabanına bağlantı yapılacaksa
	onun adı yazılır.

• MySQL sunucuya bağlanma kodunu bir kütüphane şeklinde oluşturacağız. Çünkü bundan sonraki tüm örneklerde her seferinde bağlantı kodları yazmak istemiyoruz. Aşağıda verilen MySQL bağlantı kodunu **mysqlbaglan.php** adıyla "C:\wamp\www" klasörüne kaydediniz.



Kod 14.2. PHP ile MySQL veri tabanı bağlantısı

• Yukarıdaki örnekte eğer mysqli_connect ile veri tabanına bağlantı başarısız olursa \$baglanti değişkeni değeri FALSE olacaktır, bu durumda if bloğu çalışacak ve mysqli_connect_error() fonksiyonu sayesinde hatanın sebebi ekranda kullanıcıya bildirilecektir.

http://localhost/mysqlbaglan.php

• Yukarıdaki URL ile kodu çalıştırdığınızda ekranda boş bir sayfa oluşmalıdır. Herhangi bir hata mesajı <u>görüntülenmiyorsa</u> bağlantı sorunsuz olarak gerçekleşmiştir. Hata varsa hatanın sebebi de ekranda görüntülenecektir.

(14.7) PHP İLE MYSQL VERİ TABANINA BİLGİ GİRMEK

• Bölüm 14.4'de oluşturduğumuz personel veri tabanında bulunan ad, soyad, birim ve maas alanlarına bilgi girmek için öncelikle **kayitformu.php** adıyla bir HTML form oluşturacağız.

Dikkat *Sco PHP uzantılı olup içinde hiç PHP kodu olmayan belge olabilirmi?* Kod <u>14.3'de</u> görülen uygulama sadece HTML kodlardan oluştuğu için HTML uzantılı olarak da oluşturulabilirdi, fakat ileride içine PHP kodları eklemeniz gerebileceğinden PHP uzantısı ile oluşturulmuştur.

• Kayıt formu aldığı bilgileri **kaydet.php** adlı PHP dosyaya gönderecek, bu dosya ise MySQL sunucuya kaydedecektir. Böylece 3 katmanlı mimariye uygun bir tasarım gerçekleştirilecektir.

Dikkat Geo Otomatik artan tablo alanlarına veri girmeyiniz.

memur, id alanına form kullanarak değer girmeyiniz, çünkü bu alanın değeri <u>veritaba-</u> nında **auto_increment** (otomatik artan) olarak belirlenmiştir. Form ile bilgi girmeye kalkışıldığında, <u>MySQL</u> sunucu bilgi girişine izin vermeyecek ve kodunuz hata vererek sonlanacaktır.

```
<html>
<!-- türkçe karakter desteği ayarı -->
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
<body>
  <form action="kaydet.php" method="POST">
      : <input type="text" name="ad" />
  Adı
                                             <br />
 Soyad1: <input type="text" name="soyad" /> <br />
 Birim : <input type="text" name="birim" /> <br />
 Maas : <input type="text" name="maas" /> <br />
 <input type="submit" value="KAYDET"/>
 </form>
</body>
</html>
```

Kod 14.3. Memurlar tablosuna bilgi girişi için gereken form

• Kod 14.3'ü "c:\wamp\www" klasörüne kayitformu.php adıyla kaydediniz.

 Kod 14.4'ü "c:\wamp\www" klasörüne kaydet.php adıyla kaydediniz. Kodun içinde öncelikle Bölüm 14.5'de oluşturduğumuz veri tabanı bağlantı kütüphanesi ekleniyor, sonrasında formdan gelen bilgiler alınıyor ve ekrana yazdırılıyor, ardından veri tabanına kayıt eklemek için INSERT INTO sorgusu formdan gelen veriler ile hazırlanıyor.

```
<?php
//mysql baglanti kodunu ekliyoruz
include("mysqlbaglan.php");
//degiskenleri formdan aliyoruz
extract($ POST);
echo "<html>";
//türkce karakterleri düzgün görüntüleyebilmek için eklenmiştir.
echo "<meta http-equiv='Content-Type' ";</pre>
echo "content='text/html; charset=UTF-8' />";
echo "Girdiginiz bilgiler:</br>";
echo "Adi
           :$ad </br>";
echo "Soyadi:$soyad</br>";
echo "Birim :$birim</br>";
echo "Maas :$maas</br>";
//sorguyu hazirliyoruz
$sql = "INSERT INTO memurlar ".
       "(ad, soyad, birim, maas) ".
       "VALUES ( '$ad', '$soyad', '$birim', '$maas')";
//sorguyu veritabanina gönderiyoruz.
$cevap = mysqli query( $baglanti,$sql);
//eger cevap FALSE ise hata yazdiriyoruz.
if(!$cevap) {
    echo '<br>Hata:' . mysqli error($baglanti);
else
    echo "Veritabanina eklendi.";
    echo "<br/>><a href='anasayfa.php'> Geri Dön</a>";
echo "</html>";
//veritabani baglantisini kapatiyoruz.
mysqli close($baglanti);
```

INSERT INTO sorgusu mysqli query() fonksiyonu ile veri tabanına gönderiliyor. Alınan cevaba göre ekrana hata mesajı veya anasayfaya dönüş linki gösteriliyor. anasayfa.php dosyası konunun sonunda oluşturacaktır. En son satırda ise mysqli close() ile veri tabanı bağlantısı kapatılıyor.

```
http://localhost/kayitformu.php
```

Yukarıdaki adresi kullanarak kayitformu.php çalıştırıp Ali KALE adlı bir Muhasebe memuru oluşturup bu memura 2500 TL ٠ maaş girdiğimizde kod hatasız çalıştıysa phpMyAdmin'i açıp personel veri tabanında memurlar tablosunda eklenmiş olan kaydı Şekil 14.6.1'deki gibi görebiliyor olmalısınız.



Şekil 14.10. phpMyAdmin programında tablo kayıtlarını görmek.

maas

Disa Aktar

(14.8) PHP İLE MYSQL VERİ TABANINDAKİ TABLOLARDAN BİLGİ OKUMAK

• Bu bölümde, 14.6 bölümünde bilgi girişi yaptığımız Personel veri tabanındaki memurlar tablosu içeriğini ekrana yazdıracağız. Diğer bir ifade ile mysql sunucudan veri çekip ekranda görüntüleyeceğiz. Bunun için aşağıdaki **listele.php** kodunu kullanacağız.



Kod 14.5. Memurlar tablosundan veri çekip ekranda tablo olarak görüntüleme

```
<?php
//mysql baglanti kodunu ekliyoruz
include("mysqlbaglan.php");</pre>
```

```
//sorguyu hazirliyoruz
$sql = "SELECT * FROM memurlar";
```

```
//sorguyu veritabanina gönderiyoruz.
$cevap = mysqli query($baglanti,$sql);
```

```
//eger cevap FALSE ise hata
yazdiriyoruz.
if(!$cevap){
    echo '<br>Hata:'.
mysqli_error($baglanti);
}
```

```
//sorgudan gelen tüm kayitlari tablo
içinde yazdiriyoruz.
//önce tablo başlıkları oluşturalım
echo "<html>";
```

```
//türkçe karakterleri düzgün
görüntüleyebilmek için eklenmiştir.
echo "<meta http-equiv='Content-Type' ";
echo "content='text/html; charset=UTF-8'
/>";
echo "";
echo ">";
```

```
echo "";
echo "Memur ID";
echo "Adi";
echo "Soyadi";
echo "Birimi";
echo "Maaşi";
echo "
```

```
//veritabanından gelen cevabı satır
satır alıyoruz.
while($gelen=mysqli_fetch_array($cevap))
```

Devamı sonraki sayfada

```
// veritabanından gelen değerlerle tablo satırları
oluşturalım
   echo "".$gelen['memur id']."";
   echo "".$gelen['ad']."";
   echo "".$gelen['soyad']."";
   echo "".$gelen['birim']."";
   echo "".$gelen['maas']."";
// tablo kodunu bitirelim.
echo "";
echo "<br/>><a href='anasayfa.php'> Geri Dön</a>";
echo "</html>";
//veritabani baglantisini kapatiyoruz.
mysqli close($baglanti);
?>
```

• Yukarıdaki kodu "c:\wamp\www" klasörüne listele.php adıyla kaydediniz. Kodu örneği inceleyecek olursak, mysqli_query() fonksiyonu ile veri tabanına \$sql değişkeninde bulunan sorguyu gönderiliyor ve gelen kayıt listesi "\$cevap" değişkenine atanıyor.

Dikkat Ger SQL dili ile kayıt listeleme için SELECT kullanımı

SELECT komutu bir tablodaki belirli alanların tüm satırlarının listelenmesini sağlar. Eğer tablodaki tüm alanlar listelenecekse alan adı olarak * girilebilir. Örneğin, memurlar tablosundaki tüm kayıtları listelemek için aşağıdaki sorgu kullanılır.

SELECT * FROM memurlar;

Eğer sadece ad ve maaş listelenecekse

SELECT ad, maas FROM memurlar;

şeklinde kullanılabilir.

• Sonraki satırlarda mysql_fetch_array() fonksiyonu yardımı ile sorgudan dönen \$cevap değişkenini diziye atmak için kullanıyoruz. while döngüsü dizide kayıt bitinceye kadar devam etmektedir. Döngünün her bir dönüşünde 1 satır veri tabanı kaydı elde edilmektedir. Bu kayıttaki tüm alanlar tek tek alınarak bir tablo satırı haline getirilerek ekrana yazdırılıyor. Son satırda ise veri tabanı bağlantısı kapatılıyor. Kod çalıştırıldığında Şekil 14.11'deki gibi bir görüntü oluşur. Memurlar tablosunda 3 adet memur kaydı bulunmaktadır.



Şekil 14.11. MySQL veritabanından bilgi okuma örneği

(14.9) PHP İLE MYSQL VERİ TABANINDAKİ TABLOLARDAN BILGİ SILME

- Bu bölümde, 14.6 bölümünde bilgi girişi yaptığımız Personel veri tabanındaki memurlar tablosundan satır silme işlemi gerçekleştireceğiz. Bunun için SQL diline ait DELETE FROM sorgusunu kullanacağız.
- DELETE FROM sorgusunu kullanabilmek için öncelikle hangi memur_id değerine sahip kayıdın silineceğinin bilinmesi gereklidir. Bunun için kullanıcıya bir kayıt listesi görüntüleyip kullanıcının bu listeden memur_id numaralarını görmesini ve istediği birini tıklayarak seçmesini sağlayacağız.

Dikkat & SQL dili ile kayıt silme için DELETE FROM kullanımı
DELETE FROM sorgusu WHERE koşul yapısı aracılığıyla silinecek kayıta ait birincil anahtar değerini alır ve ilgili tablodan satırı bulup siler. Örneğin memurlar tablosundan memur id numarası 3 olan kaydı silmek için
DELETE FROM memurlar WHERE memur id=3;
şeklinde bir sorgu yazılmalıdır.

• Kullanıcıya liste görüntülemek için bölüm 14.8'de verdiğimiz örneğe ufak bir ekleme yapacağız. Bu ekleme ile kayıtların listelendiği tablonun her satırında bir SİL linki oluşacaktır. Aşağıda buna ait **silmelistesi.php** dosyası kodu görülmektedir.

```
<?php
//mysql baglanti kodunu ekliyoruz
include("mysqlbaglan.php");</pre>
```

```
//sorguyu hazirliyoruz
$sql = "SELECT * FROM memurlar";
```

```
//sorguyu veritabanina gönderiyoruz.
$cevap = mysqli_query($baglanti,$sql);
```

```
//eger cevap FALSE ise hata yazdiriyoruz.
if(!$cevap ){
    echo '<br>Hata:' . mysqli_error($baglanti);
```

```
//sorgudan gelen tüm kayitlari tablo içinde yazdiriyoruz.
//önce tablo başlıkları oluşturalım
echo "<html>";
```

```
//türkçe karakterleri düzgün görüntüleyebilmek için eklenmiştir.
echo "<meta http-equiv='Content-Type' ";
echo "content='text/html; charset=UTF-8' />";
```

```
echo "";
echo "";
echo ">Memur ID";
echo ">Adi";
echo ">Soyadi";
echo ">Birimi";
echo ">Maaşi";
echo ">Maaşi";
echo "
```

//veritabanından gelen cevabı satır satır alıyoruz.
while(\$gelen=mysqli_fetch_array(\$cevap))

```
// veritabanından gelen değerlerle tablo satırları
//oluşturalım
echo "".$gelen['memur_id']."";
echo "".$gelen['ad']."";
echo "".$gelen['soyad']."";
echo "".$gelen['birim']."";
echo "".$gelen['birim']."";
//sil linki oluşturalım.
echo "".$gelen['maas']."";
//sil linki oluşturalım.
echo "".$gelen['memur_id'];
echo $gelen['memur_id'];
echo ">Sil</a>";
}
// tablo kodunu bitirelim.
echo "";
echo "";
echo "";
echo "";
echo "";
```

```
//veritabani baglantisini kapatiyoruz.
mysqli_close($baglanti);
?>
```

Kod 14.6. Memurlar tablosundan kayıt silme için silme listesi görüntüleme

• Kod 14.6'yı "c:\wamp\www" klasörüne silmelistesi.php adıyla kaydediniz. Bu uygulamada "Sil" linkini oluşturan satırı detaylı inceleyelim. Aşağıdaki satırda <a href=kayitsil.php?id=şeklinde bir ifadeye memur_id değeri ekleniyor.

Sil

 Bu kod parçası sayesinde oluşan SİL linki sayesinde, kullanıcı hangi satırdaki SİL linkini tıklarsa o satıra ait memur_id değerini elde edeceğiz. Bu değeri, kayıtları silme işlemini yapacak olan kayıtsil.php dosyasına URL arkasına ekleme şeklinde aktaracağız. Örneğin kullanıcı 3 nolu kayıttaki SİL linkini tıklarsa http://localhost/kayitsil.php?id=3 şeklinde bir URL oluşacak ve kullanıcı bu URL'ye yönlenecektir. Bu durum Şekil 14.12'de görülmektedir.

W http://lo	calhos	.il_listesi.php >	+		-		×
🗲 🛞 locall	nost/sil_l	istesi.p 🔻 Cl	Q Search		>>		≡
MySQL sum	icuya b	aglanti kuruld	u				
Memur ID	Adı	Soyadı	Birimi	Maaşı			
1	Ali	KALE	Muhasebe	2500	Sil		
3	Selim	AKSAKAL	Bilgi Islem	5000	S1)	
4	Murat	TANMAN	pazarlama	4000	SI		
localhost/kayits	il.php?id	=3					

Şekil 14.12. Silinecek kaydın seçilmesi

• Şimdi de kendisine gelen memur_id parametresini PHP ile alıp DELETE FROM sorgusunu oluşturan kayitsil.php dosyasını oluşturalım.

```
<?php
//mysql baglanti kodunu ekliyoruz
include("mysqlbaglan.php");</pre>
```

```
//sorguyu hazirliyoruz
$sql = "DELETE FROM memurlar WHERE memur_id=".$_GET['id'];
```

```
//sorguyu veritabanina gönderiyoruz.
$cevap = mysqli query($baglanti,$sql);
```

```
echo "<html>";
//türkçe karakterleri düzgün görüntüleyebilmek için
eklenmiştir.
echo "<meta http-equiv='Content-Type' ";</pre>
```

```
echo "content='text/html; charset=UTF-8' />";
```

```
if(!$cevap ){
    echo '<br>Hata:' . mysqli_error($baglanti);
```

```
else
```

```
echo "Kayıt Silindi!</br>";
echo " <a href='silmelistesi.php'> Listele</a>\n";
```

```
echo "</html>";
//veritabani baglantisini kapatiyoruz.
mysqli_close($baglanti);
?>
```

```
Yandaki kodu "c:\wamp\www" klasörüne
 kayitsil.php
              adıyla
                        kaydediniz.
                                      Bu
 uygulamada, URL aracılığıyla gelen id
 parametresi $ GET['id'] ile yakalanıyor
 ve DELETE FROM sorgusuna ekleniyor.
 Böylece kullanıcının seçtiği kaydın silinmesi
 sağlanıyor. Kayıt silmede problem olursa
 $cevap değişkeni FALSE olacağından
 ekrana hata mesajı gönderiliyor, eğer başarıyla
 silinmiş ise kullanıcı bilgilendiriliyor ve
 değişikliği
               görmesi
                           için
                                    tekrar
silmelistesi.php dosyasına yönlendiriliyor.
```



Şekil 14.13. SİL butonu tıklanması sonucu kayıt silindi bilgisi

http://lo	ocalhos	.il_listesi.php	× +				
🗲 🕙 locall	nost/sil_l	istesi.php 🔍	C ^e Q Se	arch		»	≡
MySQL sum	icuya b	aglanti kurulo	du.				
Memur ID	Adı	Soyadı	Birimi	Maaşı			
1	Ali	KALE	Muhasebe	2500	Sil		
4	Murat	TANMAN	pazarlama	4000	Sil		



SİL butonuna tıklanması sonucu kayıt siliniyor (Şekil 14.13) ve son durumu görmek için listele linki tıklanınca
 3 nolu kaydın bulunmadığı Şekil 14.14'te görülüyor.

(14.10) PHP İLE MYSQL VERİ TABANINDAKİ TABLOLARDA BİLGİ GÜNCELLEME

• Bu bölümde, 14.6 bölümünde bilgi girişi yaptığımız Personel veri tabanındaki memurlar tablosunda kayıt güncelleme işlemi gerçekleştireceğiz. Bunun için SQL diline ait UPDATE sorgusunu kullanacağız.

Dikkat ↔ SQL dili ile kayıt güncelleme için UPDATE kullanımı
UPDATE sorgusu WHERE koşul yapısı aracılığıyla güncellenecek kayıta ait birincil anahtar değerini alır. Böylece, ilgili satırı bularak SET anahtar kelimesi ile belirtilen alan adı = değer çiftlerindeki değerlere göre alanları günceller. Örneğin memurlar tablosundan memur_id numarası 3 olan kaydın adını Ece, soyadını Er, birimini muhasebe, maaşını 4000 tl olarak güncellemek için
UPDATE memurlar SET
ad='Ece',
soyad='Er',
birim='Muhasebe',
maas='4000'
WHERE memur_id=1
şeklinde bir sorgu yazılmalıdır.

- UPDATE sorgusunu kullanabilmek için öncelikle hangi memur_id değerine sahip kaydın güncelleneceğinin bilinmesi gereklidir. Bunun için kullanıcıya bir kayıt listesi görüntüleyip kullanıcının bu listeden memur_id numaralarını görmesini ve istediği birini tıklayarak seçmesini sağlayacağız.
- Kullanıcıya liste görüntülemek için Kod 14.6'da verdiğimiz örnekte ufak bir değişiklik yapacağız. Bu değişiklik ile kayıtların listelendiği tablonun her satırında bir "Güncelle" linki oluşacaktır. Aşağıda buna ait güncellelistesi.php dosyası kodu görülmektedir.

```
//mysql baglanti kodunu ekliyoruz
include("mysqlbaglan.php");
//sorquyu hazirliyoruz
$sql = "SELECT * FROM memurlar";
//sorquyu veritabanina gönderiyoruz.
$cevap = mysqli query($baglanti,$sql);
//eger cevap FALSE ise hata yazdiriyoruz.
if(!$cevap ) {
    echo '<br>Hata:' .
mysqli error($baglanti);
//sorgudan gelen tüm kayitlari tablo içinde
yazdiriyoruz.
//önce tablo başlıkları oluşturalım
echo "<html>";
//türkçe karakterleri düzgün görüntüleyebilmek
icin eklenmistir.
echo "<meta http-equiv='Content-Type' ";</pre>
echo "content='text/html; charset=UTF-8' />";
```

<?php

```
echo "";
echo "";
echo ">Memur ID";
echo ">Adı";
echo ">Soyadı";
echo ">Birimi";
echo ">Birimi";
echo ">Maaşı";
echo "
```

//veritabanından gelen cevabı satır satır alıyoruz.
while(\$gelen=mysqli_fetch_array(\$cevap))

```
// veritabanından gelenlerle tablo satırları oluşturalım
echo "".$gelen['memur_id']."";
echo "".$gelen['ad']."";
echo "".$gelen['soyad']."";
echo "".$gelen['birim']."";
echo "".$gelen['maas']."";
echo "".$gelen['maas']."";
echo "<a href=guncelle.php?id=";
echo $gelen['memur_id'];
echo ">Güncelle</a>
```

```
echo "";
echo "<br/>dr/><a href='anasayfa.php'> Geri Dön</a>";
echo "</html>";
```

```
//veritabani baglantisini kapatiyoruz.
mysqli_close($baglanti);
?>
```

 Yukarıdaki kodu "c:\wamp\www" klasörüne güncellelistesi.php adıyla kaydediniz. Bu uygulamada "Güncelle" linkini oluşturan satırı detaylı inceleyelim. Aşağıdaki satırda <a href=guncelle.php?id= şeklinde bir ifadeye memur_id değeri ekleniyor.

Güncelle

Bu kod parçası sayesinde oluşan "Güncelle" linki sayesinde, kullanıcı hangi satırdaki "Güncelle" linkini tıklarsa o satıra ait memur_id değerini elde edeceğiz. Bu değeri, kayıtları güncelleme formu olan guncelle.php dosyasına URL arkasına ekleme şeklinde aktaracağız. Örneğin kullanıcı 3 nolu kayıttaki "Güncelle" linkini tıklarsa http://localhost/guncelle.php?id=3 şeklinde bir URL oluşacak ve kullanıcı bu URL'ye yönlenecektir. Bu durum Şekil 14.15'de görülmektedir.



Şekil 14.15. Güncellenecek kaydın seçilmesi

• İkinci aşama olarak kendisine gelen memur_id parametresini alıp bu kaydı bulan ve bunu bir kayıt formunun input elemanları içine yerleştiren PHP kodunu oluşturalım. Kullanıcı, seçtiği değerlerin hazır olarak geldiği input elemanlarındaki değerleri dilerse değiştirebilir, değiştirmezse hazır gelen değerler ile bilgiler veri tabanında güncellenir.

```
<?php
//mysql baglanti kodunu ekliyoruz
include("mysqlbaglan.php");</pre>
```

```
//sorguyu hazirliyoruz
$sql = "SELECT * FROM memurlar WHERE
memur_id=".$_GET['id'];
```

```
//sorguyu veritabanina gönderiyoruz.
$cevap = mysqli query($baglanti,$sql);
```

```
//eger cevap FALSE ise hata yazdiriyoruz.
if(!$cevap ){
    echo '<br>Hata:' . mysqli_error($baglanti);
}
```

```
//veritabanından gelen cevabı alıyoruz.
$gelen=mysqli_fetch_array($cevap);
?>
```

```
→ <html>
```

```
<!-- türkçe karakter desteği ayarı -->
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
<body>
<form action="guncelkaydet.php?id=<? echo
$ GET['id'] ?>"
method="POST">
Adı:
<input type="text" name="ad" value="<? echo</pre>
$gelen['ad']?>" /> <br />
Soyad1:
<input type="text" name="soyad" value="<? echo</pre>
$gelen['soyad'] ?>" /> <br />
Birim :<input type="text" name="birim"</pre>
value="<? echo $gelen['birim'] ?>" /> <br />
Maaş : <input type="text" name="maas"</pre>
value="<? echo $gelen['maas'] ?>" /> <br />
<input type="submit" value="KAYDET"/>
</form>
</body>
</html>
```

• Yukarıdaki kodu "c:\wamp\www" klasörüne güncelle.php adıyla kaydediniz. Bu kod kendisine gelen memur_id parametresini SELECT sorgusuna parametre olarak vermektedir. Bu sayede aranan kişinin bilgileri veri tabanından getirilmektedir. Veri tabanından gelen değer \$gelen dizisi içinde depolanmıştır. Bu dizinin her bir öğesi INPUT etiketlerinin VALUE niteliklerine yazdırılmıştır. Bu sayede kullanıcılar HTML formun INPUT elemanları içinde kendi seçtikleri kaydın verilerini göreceklerdir. Ayrıca FORM etiketinin ACTION niteliğinde aşağıdaki gibi guncelkaydet.php dosyasına memur_id değeri iletilmektedir. Bu id değeri UPDATE sorgusunda kullanılacaktır.

guncelkaydet.php?id=<? echo \$_GET['id'] ?>

Şekil 14.15'deki 7 numaralı kaydın sağındaki "Güncelle" linki tıklanarak **güncelle.php** dosyası çalıştırıldığında Şekil 14.16'daki gibi bir görüntü elde edilir.

Bu aşamada kullanıcı günceleme işlemini gerçekleştirip "KAYDET" butonunu tıkladığında bu kaydın memur_id parametresini alınıp UPDATE sorgusunu oluşturan guncelkaydet.php dosyasına gönderilecektir.

Adı : Ali	
Soyadı: KALE	
Birim : Bilgi İşlem	
Maaş : 5000	
KAYDET	

Şekil 14.16. Güncellenecek kaydın bilgilerini içeren form

37

```
<?php
//mysql baglanti kodunu ekliyoruz
include("mysqlbaglan.php");
//degiskenleri formdan aliyoruz
extract($ POST);
//sorquyu hazirliyoruz
$sql ="UPDATE memurlar ".
      "SET
ad='$ad', soyad='$soyad', birim='$biri
m',maas='$maas' ".
      "WHERE memur id=".$ GET['id'];
//sorguyu veritabanina gönderiyoruz.
$cevap = mysqli query(
$baglanti,$sql);
```

```
echo "<html>";
//türkçe karakterleri düzgün görüntüleyebilmek
icin eklenmistir.
echo "<meta http-equiv='Content-Type' ";</pre>
echo "content='text/html; charset=UTF-8' />";
//eqer cevap FALSE ise hata yazdiriyoruz.
if(!$cevap){
    echo '<br>Hata:' . mysqli error($baglanti);
else{
    echo "Kayıt Güncellendi.";
    echo "<br/>><a href='anasayfa.php'> Geri
Dön</a>";
echo "</html>";
//veritabani baglantisini kapatiyoruz.
mysqli close($baglanti);
?>
```

• Yukarıdaki kodu "c:\wamp\www" klasörüne **guncelkaydet.php** adıyla kaydediniz. Bu uygulamada, URL aracılığıyla gelen id parametresi \$_GET['id'] ile yakalanıyor ve UPDATE sorgusuna ekleniyor. Böylece, kullanıcının seçtiği kayda ait id değeri ve güncelleme formundan gelen bilgiler ile bir UPDATE sorgusu oluşturularak veri tabanı sunucusuna gönderiliyor.



Şekil 14.17. Güncelle butonu tıklanması sonucu kayıt güncellendi bilgisi

Kod 14.11. Personel Otomasyonu programı ana menüsü

```
<html>
<!-- türkçe karakter desteği ayarı -->
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
<body>
<h1> Personel Otomasyonu </h1>
<a href="kayitformu.php"> Kayıt Gir </a>
<a href="listele.php"> Kayıtları Listele </a>
<a href="silmelistesi.php"> Kayıt Sil </a>
<a href="guncellelistesi.php"> Kayıt Güncelle </a>
\langle ul \rangle
</body>
</html>
```

- Yukarıdaki kodu "c:\wamp\www" klasörüne anasayfa.php adıyla kaydediniz. Bu kod aracılığıyla
 - ➢ veri tabanına kayıt ekleme,
 - ▹ kayıt listeleme,
 - ➤ kayıt silme,
 - ➢ kayıt güncelleme

işlemlerini gerçekleştirebilirsiniz. Bu örnek uygulama ile birlikte PHP ile MySQL bağlantıları bölümü sona ermektedir. Uygulama kodlarını kısa tutabilmek adına tasarımda CSS öğeleri kullanılmamıştır. Siz, kendi zevkinize göre harici CSS dosyaları oluşturarak bu kodlara ekleyebilirsiniz. Böylece estetik açıdan daha gösterişli ve ergonomik tasarımlar elde edebilirsiniz.

• Internet sitelerinden veya farklı kitaplardan daha kapsamlı örnekleri inceleyerek, kodlama pratiğinizi arttırabilir ve daha gelişmiş uygulamalar oluşturabilirsiniz.