

T.C.  
MİLLÎ EĞİTİM BAKANLIĞI



# MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN  
GÜÇLENDİRİLMESİ PROJESİ)

ENDÜSTRİYEL OTOMASYON  
TEKNOLOJİLERİ

İNTERNET PROGRAMCILIĞI  
UYGULAMALARI

Ankara, 2009

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, mesleki ve teknik eğitim okul ve kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğrenme materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere mesleki ve teknik eğitim okul ve kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlık'ta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

# İÇİNDEKİLER

AÇIKLAMALAR .....	iii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ-1 .....	3
1. VERİ TABANI BAĞLANTISI .....	3
1.1. PostgreSQL Ara Yüzünün Ana Hatları .....	3
1.2. PHP'den PostgreSQL'e Bağlantı Çeşitleri.....	4
1.2.1.Resource Pg_Connect (String Bağlantı İfadeleri).....	4
1.2.2. Bool Pg_Close (Resource Connection).....	5
1.2.3. Resource Pg_Pconnect (String Bağlantı İfadesi.....	5
1.2.4. Resource Pg_Query (Resource Bağlantı, String Sorgu) .....	5
1.2.5.String Pg_ErrorMessage (Resource Bağlantı).....	5
1.2.6.Int Pg_Freeresult (Resource Result) .....	6
1.2.7.Int Pg_Numrows (Resource Result).....	6
1.2.8.Int Pg_Numfields (Resource Result).....	6
1.2.9.String Pg_Fieldname (Resource Result- Int Col).....	6
1.2.10.Mixed Pg_Result (Resource Result- Int Row- Mixed Col) .....	6
1.2.11. Array Pg_Fetch_Array [Resource Result- Int Row ( Int Result_Type)] ...	7
1.2.12.Object Pg_Fetch_Object[Resource Result- Int Row ( Int Result_Type)]...7	
1.3. Apache + PHP + PostgreSQL .....	8
1.4. PostgreSQL Ara Yüzünü.....	11
1.4.1.Veritabanı Tarayıcısı Yapımı.....	11
UYGULAMA FAALİYETİ.....	20
ÖLÇME VE DEĞERLENDİRME.....	21
ÖĞRENME FAALİYETİ-2 .....	22
2. PHP İLE VERİ TABANI SORGULAMA.....	22
2.1. Veri Girişi.....	22
2.2. Veri Listeleme .....	25
2.2.1. Alfabetik Olarak Veri Listeleme .....	27
2.2.2. Verileri Tersten Listeleme.....	29
2.3. Veri Arama.....	30
UYGULAMA FAALİYETİ.....	33
ÖLÇME VE DEĞERLENDİRME.....	34
ÖĞRENME FAALİYETİ-3 .....	35
3. PHP İLE VERİ GÜNCELLEME VE SİLME.....	35
3.1. Veri Güncelleme .....	35
3.2. Veri Silme.....	37
UYGULAMA FAALİYETİ.....	39
ÖLÇME VE DEĞERLENDİRME.....	40
MODÜL DEĞERLENDİRME .....	41
CEVAP ANAHTARLARI.....	42
KAYNAKÇA .....	43

# AÇIKLAMALAR

<b>KOD</b>	<b>481BB0086</b>
<b>ALAN</b>	<b>Endüstriyel Otomasyon Teknolojileri</b>
<b>DAL/MESLEK</b>	<b>Endüstriyel Kontrol Teknisyenliği</b>
<b>MODÜLÜN ADI</b>	<b>İnternet Programcılığı Uygulamaları</b>
<b>MODÜLÜN TANIMI</b>	<b>İnternet programcılığı uygulamalarının anlatıldığı öğrenme materyalidir.</b>
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	Veri Tabanı modülünü almış olmak
<b>YETERLİK</b>	İnternet program uygulaması yapmak
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Veri tabanı uygulamalarını hatasız yapabileceksiniz. <b>Amaçlar</b> <ol style="list-style-type: none"><li>1. Betik dili ile veri tabanı bağlantısını doğru olarak yapabileceksiniz.</li><li>2. Betik dili ile veri tabanı sorgulamasını sorunsuz bir şekilde yapabileceksiniz.</li><li>3. Betik dili veri tabanına veri ekleme ve silme işlemlerini hatasız bir şekilde yapabileceksiniz.</li></ol>
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Bilgisayar laboratuvarı <b>Donanım:</b> Bilgisayar
<b>ÖLÇME VE DEĞERLENDİRME</b>	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

# GİRİŞ

## Sevgili Öğrenci,

Teknoloji ve bilginin hızla yenilendiği dünyamızda web iletişimi, hayatımızın vazgeçilmezlerinden olmuştur. Bilginin hızlı gelişimini takip etmek için bu iletişim sistemlerini bilmek ve uygulamak kişiyi bir adım öne geçirmekte önemli bir yeterlidir. 21.yüzyılda zamanla yarıştığımız ve yarışı kazanmak için uzaktan bilgiye erişim imkânlarından en iyi şekilde yararlanmamız gerektiği bilinen bir gerçektir.

İnternet programcılığı ve güvenliği, bilgiye erişim noktasında çok önemli bir yere sahiptir. İnternet programcılığında ise veri tabanı özellikle üzerinde durulması gereken bir alandır. Çünkü online sistemlerin bir çoğu veritabanı yönetimi üzerine kurulmuştur ve internet kullanıcılarının bu sistemde istediği bilgiye pratik bir şekilde ulaşabilmesi sistemden öncelikli istenen yeterlikler arasındadır. Bu pratikliği sağlamak için ise veri tabanındaki bilgiyi veri tabanı programları (MySQL, PostgreSQL) ve ara yüzler (PHP,ASP vb.) kullanarak sorgulama işlemini bilmek gerekir.

Bu modülle, veri tabanı programlarını kullanarak internet veri tabanı programlama alanında sorgulama yani kayıtları değişik özelliklerine göre süzme becerisine sahip olacaksınız.



# ÖĞRENME FAALİYETİ-1

## AMAÇ

Betik dili ile veri tabanı bağlantısını doğru olarak yapabileceksiniz.

## ARAŞTIRMA

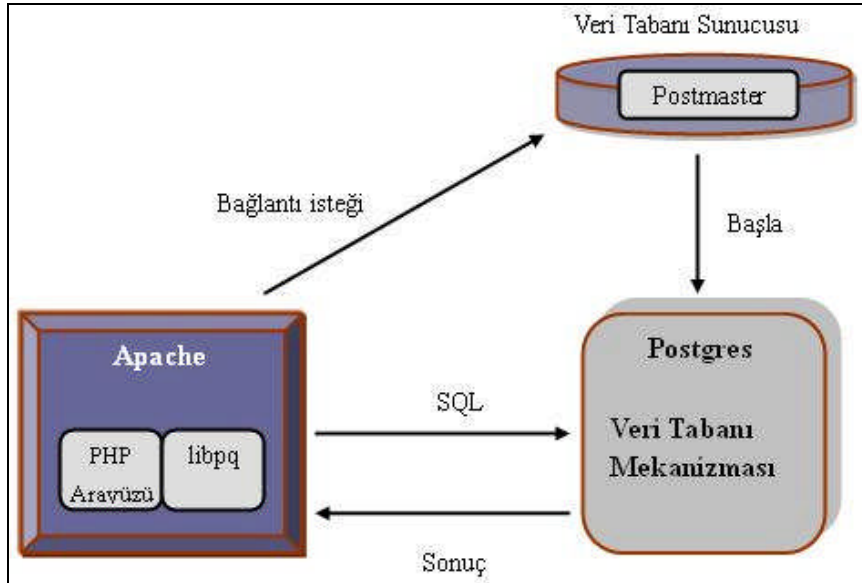
- PostgreSQL hakkında internet ve okul kütüphanesinden yararlanarak araştırma yapınız.

## 1. VERİ TABANI BAĞLANTISI

### 1.1. PostgreSQL Ara Yüzünün Ana Hatları

Şimdiye kadar PHP ve PostgreSQL'in temel kullanımını öğrendik. Bu bölümde PHP içinden PostgreSQL'e nasıl ulaşılabileceğini öğreneceğiz.

Öncelikle PHP ile PostgreSQL ilişkisinin yapısını öğrenelim. PHP içinde PostgreSQL diğer programlarda olduğu gibi C dili temellidir. PostgreSQL libpq adında C dilinde kullanılabilen bir API'ya sahiptir. PHP içinden libpq API'ı çağırılarak çalıştırılabilir. Bu yapı oldukça basit bir yapıdır.



Şekil 1.1: PHP - PostgreSQL bağlantısı

- Sistemin çalışması aşağıdaki aşamalardan meydana gelir.
- PostgreSQL bağlantısı için fonksiyon çağrılır.
- Libpg sayesinde bağlantı isteği veri tabanı sunucusuna (postmaster) yeni PostgreSQL'e gönderilir.
- Bu istek alındığında postmaster veri tabanı mekanizmasını (postgres) başlatır.
- Bu durumda PostgreSQL kullanılır hâle gelir.
- Bu bölümde SQL ifadesi şekli de sorgu iletilir.
- Sorgu postgres ile işleme konur ve sonuç döndürülür.

Veri tabanı istek mekanizması bu şekilde sürekli tekrarlanarak işlemler gerçekleştirilir.

## 1.2. PHP'den PostgreSQL'e Bağlantı Çeşitleri

PHP'den PostgreSQL'e bağlantı kurma yolları aşağıda anlatılmıştır.

### 1.2.1.Resource Pg\_Connect (String Bağlantı İfadeleri)

Pg\_connect ara yüzü PostgreSQL ile bağlantı kurmaya yarar. Dönen veri resource tipindedir. Resource PHP haricî veri yönetimidir. Resource tipin değeri eğer script içinde kullanılmaz ise serbest bırakılır. Yani veri tabanı bağlantısını kesmek bile otomatik olarak kesilir, böylelikle gereksiz bağlantılar sürdürülmez.

Pg\_connect PostgreSQL ile bağlantı hatası oluştuğunda flase değeri döndürür. İçinde kullanılan parametreler bağlantının detayları hakkında bilgi verir. Aşağıdaki bölümde kullanım formu anlatılmıştır.

Anahtar kelime = değer

Aşağıda verilen anahtar kelimeler kullanılmadığı durumda varsayılan değerler etkin olur. Ayarlamaları manuel olarak yapmak için aşağıdaki anahtar kelimeler kullanılabilir.

Anahtar Kelime	Anlamı	Varsayılan Değer
host	Postmasterin bulunduğu sunucu	Bilgisayarın kendi
dbname	Veri tabanı adı	Apache kullanıcı adı
user	PostgreSQL in kullanıcı adı	Apache kullanıcı adı
password	PostgreSQL bağlantısı için şifre	Şifre yok
port	Postmaster sunucusunun kullandığı port numarası	5432
options	PostgreSQL için ayarlar	Boş
tty	Hata ayıklama terminali	Boş

**Tablo 1.1: Argüman olarak kullanılan anahtar kelimeler**

Şimdi her bir bölümde kullanılacak anahtar kelimeleri birkaç örnek ile açıklayalım.



### ➤ Host parametresi

Bağlı bulunan host adına göre değişik işlevler yerine getirir. Örneğin postmastere aşağıdaki makinelerin bağlantı yapmaya çalıştığını düşünelim. Bu bağlantıdaki farklılıklar örnekte verilmiştir.

localhost	TCP/IP ile bağlantı
(belirtilmediği durum)	UNIX domain socket ile bağlantı

PostgreSQL bu bölümde eğer varsayılan değer kabul edilmiş ise yani herhangi atama kullanılmıŝ ise TCP/IP bağlantılar kabul edilmez. Bu, güvenlik amacı ile bu şekilde tasarlanmıştır. Sonuç olarak bu tanımlama yapılmadan localhost yerel makine bağlanmaya çalışıldığında hata oluşacaktır. Eğer diğeri hostlardan bağlantı yapılmayacaksa host parametresini kullanmak gereksizdir.

### ➤ Tty parametresi

Tanımlanması çok gerekli bir parametre değildir. Çünkü bu tanımlama PostgreSQL'in güvenlik alanı ile ilgilidir. "pg\_connect" in örnek kullanımı aşağıda verilmiştir.

```
$con = pg_connect("dbname=test user=nobody options='-c sort_mem=1024'");
```

## 1.2.2. Bool Pg\_Close (Resource Connection)

Pg\_connect ile yapılan bağlantıyı kapatmak için kullanılır. Eğer script pg\_close kullanılmadan bitirilmiş ise pg\_close işlemi otomatik olarak gerçekleştirilir.

## 1.2.3. Resource Pg\_Pconnect (String Bağlantı İfadesi)

Pg\_connect'e benzer. Farklılıkları aşağıdaki gibidir.

➤ Pg\_pconnect ile yapılan bağlantılar ısrarcı presistence bağlantı olarak ifade edilir. Bu tip bağlantıları kapatmak için pg\_close kullanılmaz.

➤ Eğer apache'nin alt işlevi durdurulursa bu bağlantı da durdurulur. PostgreSQL'le bağlantı performansı "pg\_connect"e göre daha fazladır.

## 1.2.4. Resource Pg\_Query (Resource Bağlantı, String Sorgu)

Bağlantı adı verilen PostgreSQL veri tabanında sorgu SQL ifadesini çalıştırır. Eğer SQL sorgusunun çalıştırılmasında herhangi bir hata oluşursa FALSE değeri döndürülür.

## 1.2.5.String Pg\_Errormessage (Resource Bağlantı)

Pg\_query’de hata oluştuğunda gelen hata mesajı karakter katarı olarak geri döndürülür. Bu ifade ile dönen hata mesajını kullanabiliriz.

### 1.2.6.Int Pg\_Freeresult (Resource Result)

Pg\_query ifadesinin kullandığı hafıza bölümünü serbest bırakır.

### 1.2.7.Int Pg\_Numrows (Resource Result)

Pg\_query ile çalıştırılan SELECT sorgulama ifadesi ile seçilen veri satır sayısını geri döndürür. Eğer bir hata oluşmuş ise -1 değeri döndürülür.

### 1.2.8.Int Pg\_Numfields (Resource Result)

Pg\_query ile çalıştırılan SELECT sorgulama ifadesi ile seçilen veri kolon sayısını geri döndürür. Eğer bir hata oluşmuş ise -1 değeri döndürülür.

### 1.2.9.String Pg\_Fieldname (Resource Result- Int Col)

Pg\_query ile çalıştırılan SELECT sorgulama ifadesi ile seçilen verilerin kolon sayılarını int col ile tanımlanan değişkene döndürür. Kolon indexi 0’dan başlar.

### 1.2.10.Mixed Pg\_Result (Resource Result- Int Row- Mixed Col)

Pg\_result pg\_query’nin çalışma sonuçlarını elde etmek için kullanılır. İçinde NULL değer içeren veriler hariç tüm sonuçla bir karakter katarı hâlinde döndürülür.

Aşağıda pg\_result un çalışması ile ilgili örnek verilmiştir.

```
$con = pg_connect("dbname=test");
    if ($con == false) {
        printf("We can't connect to Data Base. Reason:
%s<br>\n", $php_errormsg);
        exit;
    }
    $result = pg_query($con, "SELECT * FROM pg_user");
    if ($result == false){
        printf("pg_query failed. Reason: %s<br>\n",
        pg_errormessage($con));
        exit;
    }
    for ($i=0;$i<pg_numrows($result);$i++)
    {
        for ($j=0;$j<pg_numfields($result);$j++)
        {
            printf("%s: %s ", pg_fieldname($result, $j),
            pg_result($result,$i,$j));
        }
        echo"<br>\n";
    }
}
```

```
pg_close($con);
```

### 1.2.11. Array Pg\_Fetch\_Array [Resource Result- Int Row ( Int Result\_Type)]

Sorgulama sonucu “pg\_result”a aktarılır. Bu Sorgulama sonuçlarının her bir satırı ile işlem yapılmak istendiğinde bu ifade kullanılır. Alınan her ayrı sonuç birer satırı ifade eder.

Result\_type sorgulama sonucunun hangi tipte bir dizi hâlinde verileceğinin belirtildiği argümandır. Aşağıdaki dizi tipleri kullanılabilir.

<i>PGSQL_ASSOC</i>	<i>Dönen sonuç birleşik dizi olacak.</i>
<i>PGSQL_NUM</i>	<i>Dönen sonuç normal dizi olacak.</i>
<i>PGSQL_BOTH</i>	<i>Dönen sonuç hem birleşik hem normal dizi olacak.</i>

Result\_type belirtilmediği zaman otomatik olarak PGSQL\_BOTH tanımlanmış olur.

Aşağıda pg\_fetch\_array ile ilgili örnek program verilmiştir.

```
$con = pg_connect(“dbname=test”);
if ($con == false){
    printf(“We can’t connect to Data Base. Reason:
%s<br>\n”, $php_errormsg);
    exit;
}
$result = pg_query($con, “SELECT * FROM pg_user”);

if ($result == false){
    printf(“pg_query failed. Reason: %s<br>\n”,
        pg_errormessage($con));
    exit;
}
for ($i=0;$i<pg_numrows($result);$i++){
    $row = pg_fetch_array($result, $i, PGSQL_ASSOC);
    foreach ($row as $fname => $val){
        echo”$fname: $val”;
    }
    echo”<br>\n”;
}
pg_close($con);
```

### 1.2.12.Object Pg\_Fetch\_Object[Resource Result- Int Row ( Int Result\_Type)]

Pg\_fetch\_array ile aşağı yukarı aynı işlemi gerçekleştirir. Kullanılan argümanlarda pg\_fetch\_array ile aynıdır. Farkı, sonucun bir dizi şeklinde değil de bir nesne şeklinde dönmesidir.

Aşağıdaki örnekte `pg_fetch_object` kullanımını gösterilmektedir.

```
$con = pg_connect("dbname=test");
    if ($con == false){
        printf("We can't connect to Data Base. Reason:
%s<br>\n",                                $php_errormsg);
        exit;                               }
    $result = pg_query($con, "SELECT * FROM pg_user");
    if ($result == false){
        printf("pg_query failed. Reason: %s<br>\n",
        pg_errormessage($con));
        exit;                               }
    for ($i=0;$i<pg_numrows($result);$i++){
        $obj = pg_fetch_object($result, $i);
        for ($j=0;$j<pg_numfields($result);$j++){
            $fname = pg_fieldname($result,$j);
            printf("%s: %s ", $fname, $obj-
>$fname);                                }
        echo"<br>\n";                        }
    pg_close($con);
```

### 1.3. Apache + PHP + PostgreSQL

Bu bölümde bu üç uygulamanın senkronize bir biçimde çalışıp çalışmadığının kontrolünü yapacağız.

#### ➤ Giriş

Postgres kullanıcıları olarak giriş yapınız.

#### ➤ PostgreSQL kullanıcıları oluşturma

Aşağıdaki kullanıcıları oluşturunuz.

Nobody            Sınama amaçlı

Apache            Temelde veri tabanına tarayıcıdan ulaşmak amaçlı

Kullanıcıları oluşturmak için,

```
$ createuser nobody
    Shall the new user be allowed to create database? (y/n) y
    Shall the new user be allowed to create more new users? (y/n) n
```

```
$ createuser apache
    Shall the new user be allowed to create database? (y/n) y
    Shall the new user be allowed to create more new users? (y/n) n
```

### ➤ Veri tabanı oluşturma

```
$ createdb test
```

Oluşturulan veri tabanı içinde tablo oluşturmak için ve örnek veriler girilmesi için SQL ifadelerinden bir dosya hazırlayalım.

```
vi test.sql
```

```
drop table packages;  
create table packages (name text, version text, relese int);  
insert into packages values ('Apache','2.0.47', 10);  
insert into packages values ('PHP','4.3.3', 6);  
insert into packages values ('PostgreSQL','7.3.4', 2);  
insert into packages values ('BIND','9.2.2', 9);  
insert into packages values ('Sendmail','8.12.10', 1);  
insert into packages values ('Squid','2.5.3', 0);  
insert into packages values ('Samba','3.0.0', 15);  
grant select on packages to nobody;  
grant select on packages to apache;
```

Yukarıdaki ifadelere göre veri tabanındaki kayıtlar aşağıdaki gibi olur.

<u>name</u>	<u>version</u>	<u>Relese</u>
Apache	2.0.47	10
PHP	4.3.3	6
PostgreSQL	7.3.4	2
BIND	9.2.2	9
Sendmail	8.12.10	1
Squid	2.5.3	0
Samba	3.0.0	15

Varsayılan değer olarak tarayıcıdan veri tabanı kontrolüne izin verilmez. Bunun için veri tabanına apache kullanıcısına grant ile bu izini vermemiz gerekir.

### ➤ SQL ifadesinin girilmesi

```
$ psql tests -f tests.sql
```

Çıktınızı aşağıdaki çıktı ile karşılaştırınız.

```
bash-2.05b$ psql tests -f tests.sql  
DROP TABLE  
CREATE TABLE  
INSERT 17061 1  
INSERT 17062 1
```

```
INSERT 17063 1
INSERT 17064 1
INSERT 17065 1
INSERT 17066 1
INSERT 17067 1
GRANT
GRANT
```

### ➤ PHP script yapımı

Root kullanıcısı olarak çalışınız.

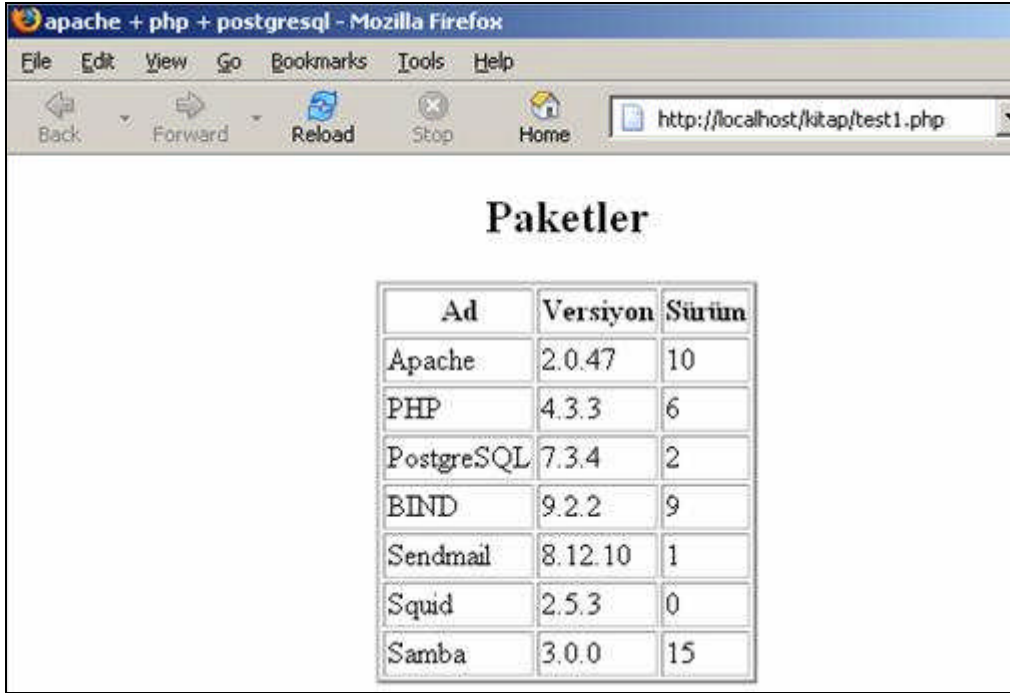
```
$ su -
vi /var/www/html/test1.php

<html>
<head>
<title>apache + php + postgresql</title>
</head>
<body>
<center>
<h2>paketler</h2>
<table border=2>
  <tr><th>name</th><th>versiyon</th><th>relese</th></tr>

<?php
  $con = pg_connect("dbname=test user=apache");
  $rtn = pg_query($con,"select * from packages");
  $num = pg_num_rows($rtn);

  for($i=0;$i<$num;$i++){
    $name = pg_fetch_result($rtn,$i,0);
    $version = pg_fetch_result($rtn,$i,1);
    $relese = pg_fetch_result($rtn,$i,2);
    echo "<tr><td>$name</td><td>$version</td><td>$relese</td></tr>";
  }
?>
</table>
</center>
</body>
</html>
```

Yukarıdaki kodların web tarayıcısındaki görüntüsü Şekil 1.2’de gösterilmiştir.



The screenshot shows a Mozilla Firefox browser window with the title 'apache + php + postgresql - Mozilla Firefox'. The address bar displays 'http://localhost/kitap/test1.php'. The main content area shows a table titled 'Paketler' with the following data:

Ad	Versiyon	Sürüm
Apache	2.0.47	10
PHP	4.3.3	6
PostgreSQL	7.3.4	2
BIND	9.2.2	9
Sendmail	8.12.10	1
Squid	2.5.3	0
Samba	3.0.0	15

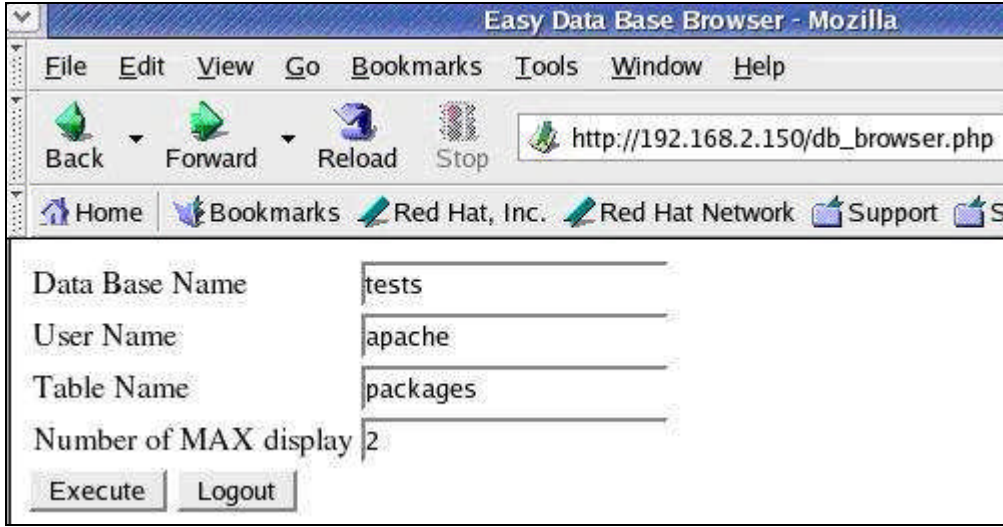
Şekil 1.2: Test1.php dosyasının web tarayıcısında gösterimi

## 1.4. PostgreSQL Ara Yüzünü Kullanma

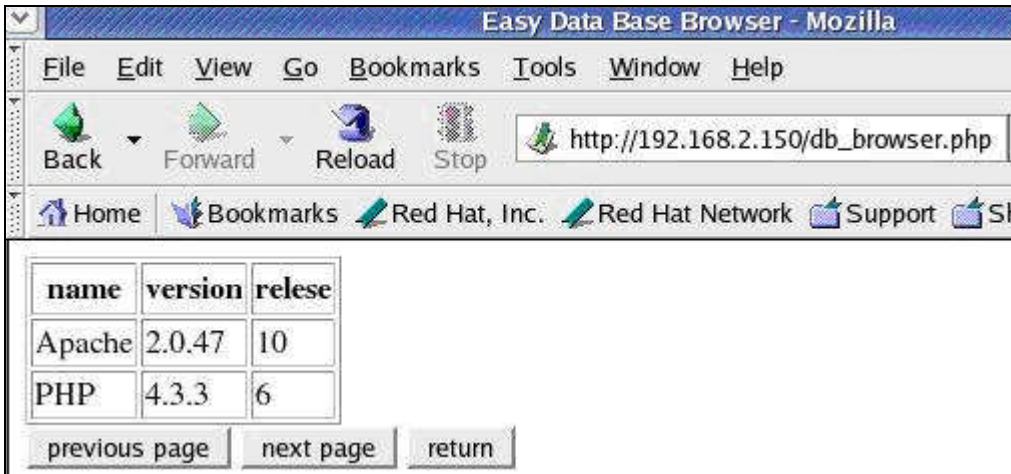
Bu bölümde PostgreSQL ara yüzünün kullanımı ile ilgili örnek verilmiştir. Burada veriler tablodan okunmakta veri tabanı tarayıcısı yaparak istenilen verilerin gösterilmesi sağlanmaktadır.

### 1.4.1. Veri Tabanı Tarayıcısı Yapımı

Aşağıdaki şekilde arama işlemine ait ara yüz gösterilmektedir. Bu bölümdeki tablolara veriler girildikten sonra execute butonuna basıldığında sonuç görüntülenmiş olur. Number of MAX display her sayfada gösterilecek maksimum satır sayısını belirtir.



Şekil 1.3 Arama işlemine ait arayüz



Şekil 1.4: Execute butonuna basıldıktan sonraki arama sonuçları

Logout düğmesine basıldığında oturum bilgileri yok edilir ve karşımıza aşağıdaki sayfa görüntüsü gelir. Bu sayfadan yine ana sayfaya dönülebilir.





Şekil 1.5: Sistemden çıkış

Veri tabanı tarayıcısı yapımı aşağıdaki işlem basamaklarından meydana gelir.

➤ **Giriş**

Root olarak giriş yapınız.

➤ **Ana scriptin yapımı (vt\_tarama.php)**

```
vi /var/www/html/vt_tarama.php
```

Veri tabanı tarama işlemi için vt\_tarama.php dosyası oluşturulduktan sonra aşağıdaki kodlar yazılır.

```
<?php 1
session_start(); 2
?> 3
<html> 4
<head> 5
<title>Easy Data Base Browser</title> 6
</head> 7
<body bgcolor=white> 8
<?php 9
if (isset($_POST[start]) || 10
isset($_POST[next]) || 11
isset($_POST[prev])) { 12
include_once(db_browser_select.inc); 13
} else if (isset($_POST[logout])) { 14
include_once(db_browser_logout.inc); 15
} else { 16
include_once(db_browser_form.inc); 17
} 18
?> 19
</body> 20
```

- 17.satır: db\_browser\_form.inc dosyası dâhil edilmektedir. Bu bölümde giriş formu gösterilmiş olur.
- 10.satır: Execute düğmesine basıldığında start değişkeni set edilir. Böylelikle daha önceden tanımlanmış \$\_POST[start] oluşturulmuş olur.
- 13.satır: db\_browser\_select.inc dâhil edilmektedir. Girilen veriler doğrultusunda işlem devreye sokulur.
- 11.- 12. satırlar: Previous page veya next page butonlarına basıldığında, değişkenlerden prev veya next set edilir. Böylelikle, \$\_POST[prev] veya \$\_POST[next] oluşturulmuş olur.
- 13 satır: db\_browser\_select.inc dâhil edilmektedir. Ve sayfa gösterim prosedürü çalıştırılmış olur.
- 14.satır: Logout düğmesine basıldığında logout değişkeni set edilmiş olur. Böylelikle \$\_POST[logout] oluşturulmuş olur.
- 15.satır: db\_browser\_logout.inc dâhil edilmektedir. Ve logout işlemi çalıştırılmış olur.

➤ **Form bölümü için script yapımı (vt\_tarama\_form.inc)**

vi /var/www/html/vt\_tarama\_form.inc

```
<?php 1
function ifset($name, $default = ) { 2
if (isset($_SESSION[$name])) 3
return($_SESSION[$name]); 4
else 5
return $default; 6
} 7
$dbname = ifset(dbname); 8
$user = ifset(user); 9
$table_name = ifset(table_name); 10
$maxline = ifset(maxline,10); 11
print<<<EOF 12
<form method=post> 13
<table> 14
```

<tr><td>Data Base Name</td><td><input type=text name=dbname	15
value=\$dbname></td></tr>	16
<tr><td>User Name</td><td><input type=text name=user value=\$user>	17
</td></tr>	18
<tr><td>Table Name</td><td><input type=text name=table_name	19
value=\$table_name></td></tr>	20
<tr><td>Number of MAX display</td><td><input type=text name=maxline	21
value=\$maxline></td></tr> </table>	22
<input type=submit name=start value=Execute>	23
<input type=submit name=logout value=Logout>	24
EOF;	25
?>	26

Bu script sadece arama kistaslarının girileceği formu gösterir.

- 3– 7. satırlar: Bu fonksiyon eğer bu değişken ilk argüman ile tanımlanmış ise ifset fonksiyonu oturum değişkenini döndürür. Eğer ikinci argüman tanımlanmamış ise (null) döndürülür.
- 8 – 11 satırlar: Bu bölümde form içinde varsayılan değer olarak gösterilecek ifadeler gösterilir. \$dbname veri tabanı adı, \$user kullanıcı adı, ve \$table\_name ise tablo adıdır. İlk anda tüm alanlara atanır. \$maxline değişkeninin varsayılan değeri ise 10'dur.
- 13 -22 satırlar: Bu bölümde form görüntüleme işlemi yapılmaktadır.

#### ➤ Çalışma durumu scripti yapımı (vt\_tarama\_sec.inc)

```
vi /var/www/html/vt_tarama_sec.inc
```

<?php	1
if (isset(\$_POST[start])) {	2
\$_SESSION[dbname] = \$_POST[dbname];	3
\$_SESSION[user] = \$_POST[user];	4
\$_SESSION[maxline] = (int)\$_POST[maxline];	5
\$_SESSION[table_name] = \$_POST[table_name];	6
\$_SESSION[offset] = 0;	7
}	8
	9
\$dbname = \$_SESSION[dbname];	10
\$user = \$_SESSION[user];	11

```

$maxline = $_SESSION[maxline];           12
$table_name = $_SESSION[table_name];     13
                                           14
if (isset($_POST[next])) {              15

    $_SESSION[offset] += $maxline;       16
} else if (isset($_POST[prev]) && $_SESSION[offset] > 0) { 17
    $_SESSION[offset] -= $maxline;       18
}                                          19
$offset = $_SESSION[offset];             20
                                           21

@$con = pg_connect(dbname=$dbname user=$user); 22
if (!$con) {                              23
    printf(User $user couldn't connect to $dbname Data Base. Reason : 24
    $php_errormsg<br>\n);                 25
    exit;                                   26
}                                           27
$table = ereg_replace(, , addslashes($table_name)); 28
$sql = SELECT * FROM $table LIMIT $maxline OFFSET $offset; 29
@$res = pg_query($con, $sql);             30

if (!$res) {                               31
    printf($sql couldn't execute. Reason : . pg_errormessage($con)); 32
    exit;                                   33
}                                           34
                                           35
$rowcnt = pg_numrows($res);               36
                                           37

print(<table border>\n);                   38
                                           39

for ($i=0; $i<$rowcnt; $i++) {            40

    print(<tr>\n);                          41

    if ($i == 0) {                          42

        $colcnt = pg_numfields($res);      43

        for ($j=0; $j<$colcnt; $j++) {     44

            printf(<th>%s</th>, pg_fieldname($res, $j)); 45
        }                                    46
    }

```

<code>print(&lt;/tr&gt;\n&lt;tr&gt;\n);</code>	47
<code>}</code>	48
<code>\$row = pg_fetch_array(\$res, \$i, PGSQL_NUM);</code>	49
<code>\$row = pg_fetch_array(\$res, \$i, PGSQL_NUM);</code>	50
<code>for (\$j=0;\$j&lt;\$colcnt;\$j++) {</code>	51
<code>print(&lt;td&gt;{\$row[\$j]}&lt;/td&gt;);</code>	52
<code>}</code>	53
<code>print(&lt;/tr&gt;\n);</code>	54
<code>}</code>	55
<code>print (&lt;/table&gt;\n);</code>	56
<code>pg_close(\$con);</code>	57
<code>pg_close(\$con);</code>	58
<code>?&gt;</code>	59
<code>?&gt;</code>	60
<code>&lt;form method=post&gt;</code>	61
<code>&lt;form method=post&gt;</code>	62
<code>&lt;input type=submit name=prev value=previous page&gt;</code>	63
<code>&lt;input type=submit name=next value=next page&gt;</code>	64
<code>&lt;input type=submit name=return value=return&gt;</code>	65
<code>&lt;/form&gt;</code>	66

Bu script veri tabanı tarayıcısı programının merkezini oluşturmaktadır. Yani veri tabanı sorgusunun ve sonuçlarının gösterildiği bölümdür. Script başlatıldığında başka bir deyişle, Execute butonuna basıldığında aşağıdaki işlemler gerçekleşir.

- Start değişkeni set edilir ( db\_browser\_form.inc 22 satır.).
- if ifadesi doğru olur (2. satır).
- 3 - 7 Satırlar: Oturum değişkenleri tanımlandığı ve ilk değer atamaları yapılan bölümdür. Temelde db\_tarama\_form.inc dosyası içinde tanımlanan değerler başlangıç değerleridir. Bununla birlikte offset değeri arama sonucu sayısı olduğu için başlangıç değeri olarak 0 atanmıştır.
- 10-13 Satırlar: Bu bölümde oturum değişkenleri atanır.
- 15- 19 Satırlar: Bu bölüm next page veya previous page seçimine göre çalıştırılır. Next page butonuna basıldığında next değişkeni set edilir.

Bununla birlikte \$maxline deęişkeni offset deęişkenine eklenir. previous page butonu seçildiğinde ise tam tersi bir işlem yapılır.

- 22 satır: Deęişkenler set edildiğinde veri tabanı bağlantısı yapılır.
- 28 Satır: Gereksiz SQL ifadeleri ereg\_replace ve addslashes kullanılarak güvenlik amacı ile girilmemiştir.
- 30 Satır: SELECT ifadesi çalıştırılır.
- 36 Satır: İşlem sonucunda satır sayısı bu bölümde ortaya çıkmaktadır.
- 38- 56 Satır: Tablo formatında görüntüleme
- 43 – 47 Satırlar: Bu kısım tablo başlıklarını göstermek için kullanılır.
- 50 – 52 Satırlar: Veriler listelenir.
- 62 – 66 Satırlar: Butonların gösterim aşaması (previous page, next page, return)

#### ➤ **Çıkış için script yapımı (db\_browser\_logout.inc)**

```
vi /var/www/html/db_browser_logout.inc
```

```
<?php 1
session_destroy(); 2

print<<<EOF 3
It completed for logout. 4
If you want login again, click<a href={$_SERVER['PHP_SELF']}>here</a>. 5
EOF; 6
?> 7
```

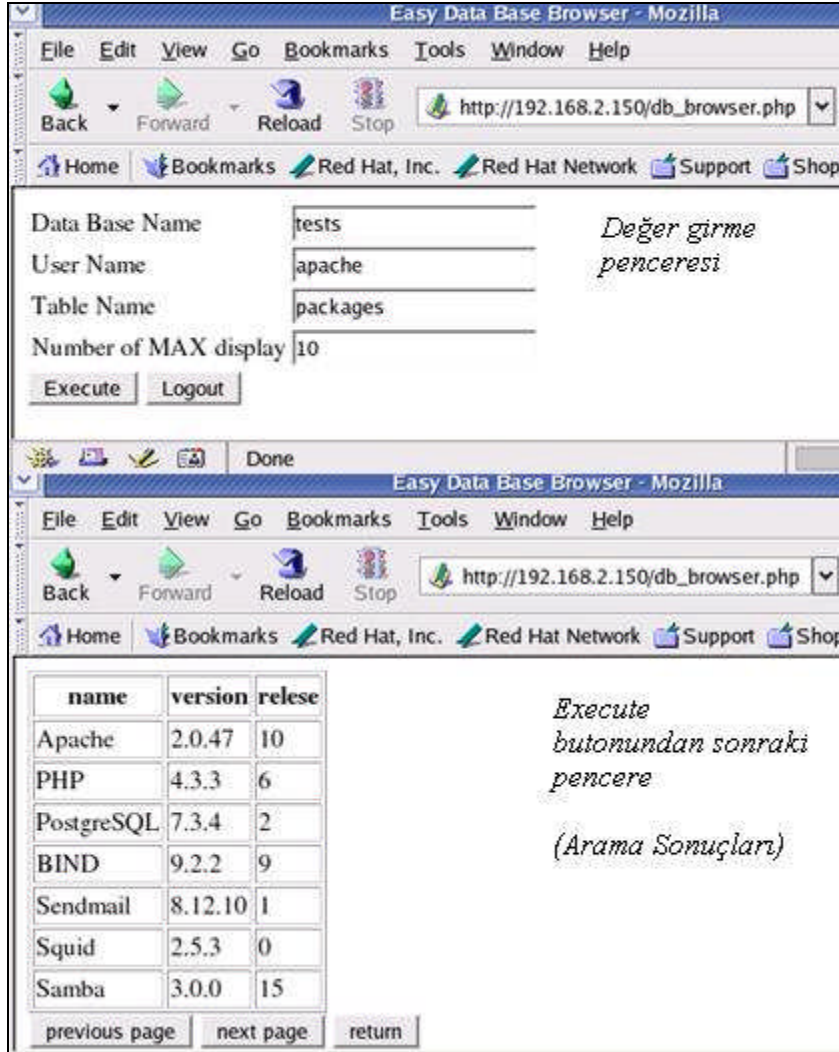
Logout butonuna basıldığında bu script çağırılır.

- 2. Satır: session\_destroy çağırılıp önceki oturum kapatılır.
- 5 Satır: Buradaki link kendi kendini çağırarak başlangıç formunun ekrana getirilmesini sağlar.

#### ➤ **Sonuçların kontrolü**

Bu bölümde, daha önce oluşturduğumuz veri tabanı kullanılmıştır. Aşağıdaki seçenekleri girerek çıktının aşağıdaki gibi olup olmadığını kontrol ediniz.

Data Base Name	tests
User Name	apache
Table Name	packages
Number of MAX display	10



Şekil 1.6: Örnek değerlerle sonuçların kontrolü

## UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarını gerçekleştiriniz.

İşlem Basamakları	Öneriler
➤ Kullanıcının veri tabanı oluşturma hakkı olan ama başka kullanıcı oluşturma hakkı bulunmayan bir PostgreSQL kullanıcısı oluşturunuz.	
➤ Bilgisayar malzemelerini girebileceğimiz urun.db adında bir veri tabanı oluşturunuz.	➤ Verileri girmek için urun.sql dosyasını oluşturmayı unutmayınız.
➤ Malzeme_adi ve adet alanlarından oluşan malzeme tablosunu oluşturunuz.	
➤ Alanlara uygun 5 adet değer giriniz.	➤ İşlemci, 54 – monitör, 60 gibi değerler giriniz.
➤ Bu değer tablosundan ilk üç satırı gösteriniz.	



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerde verilen bilgiler doğru ise (D), yanlış ise (Y)'yi işaretleyiniz.

1. Create table komutu veri tabanında tablo oluşturmak için kullanılır (D) (Y)
2. Resource pg\_connect parametresi veri tabanına bağlantı için kullanılır. (D) (Y)
3. Kullanıcıya tarayıcıdan veri tabanı kontrolüne izin verilmesi için sql dosyasında drop komutunun kullanılması gerekir.(D) (Y)

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

4. Aşağıdakilerden hangisi sorgu ifadesini çalıştırmak için kullanılan PostgreSQL ara yüzü parametresidir?

- |                        |                          |
|------------------------|--------------------------|
| A) resource pg_connect | B) bool pg_close         |
| C) resource pg_query   | D)string pg_errormessage |

5. PostgreSQL ara yüzüne bağlantı kurarken kullanılan hata ayıklama terminalini ifade eden parametre aşağıdakilerden hangisidir?

- |                |         |
|----------------|---------|
| A)errormessage | B) Tty  |
| C)password     | D) host |

### 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

Betik dili (PHP) ile veri tabanında veri sorgulama işlemlerini sorunsuz bir şekilde gerçekleştirebileceksiniz.

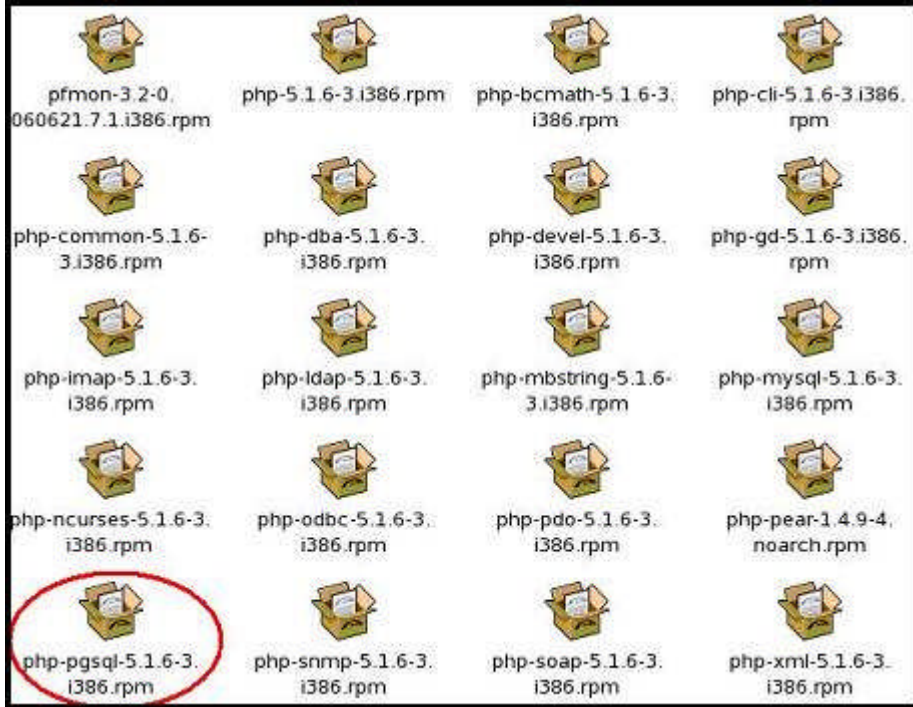
## ARAŞTIRMA

- İnternet ve çeşitli kaynaklardan, PHP dili ile veri tabanı kontrol eden programları inceleyiniz.

## 2. PHP İLE VERİ TABANI SORGULAMA

### 2.1. Veri Girişi

PHP içinde PostgreSQL'in çalışması için PHP'ye PostgreSQL modülünün eklenmesi gereklidir. Bu amaçla aşağıdaki rpm paketi kurulmuş olmalıdır. Fedora CD'si içinde mevcuttur ya da internetten indirilebilir.



Şekil 2.1: Fedora CD'sindeki PostgreSQL modülünün bulunduğu paket

“Pg\_connect” komutu içerisinde “user” ifadesinin kullanılması için öncelikle aşağıda adı verilen dosya içinde aşağıdaki bölüm değiştirilmelidir.

```
vi /var/lib/pgsql/data/pg_hba.conf
```

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
```

Postgresql’e user ile bağlantı yapabilmek için bu bölüm **trust** yapılmalıdır.

Değişiklik yapıldıktan sonra postgresQL servisi yeniden başlatılmalı.

```
#service postgres restart
```

Şimdi de postgres kullanıcısı olarak postgres içinde bir kullanıcı oluşturalım.

```
[root@bulent data]# su postgres
bash-3.1$ createuser bulentvardal
Shall the new role be a superuser? (y/n) y
CREATE ROLE
bash-3.1$ psql dosya -U bulentvardal
Welcome to psql 8.1.4, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
       \h for help with SQL commands
       \? for help with psql commands
       \g or terminate with semicolon to execute query
       \q to quit
```

```
dosya=#
```

Şimdiki basamakta “dosya” veri tabanı içinde “ogrenciler” tablosu oluşturalım.

```
dosya=# create table ogrenciler(numara integer,ad varchar(20),soyad varchar(20));
```

Kullanıcıya tablo üzerindeki tüm hakları vermek için “GRANT” komutu kullanılır. Grant komutunun kullanımı aşağıdaki gibidir.

```
GRANT izin_türü ON Tablo_adi TO Kullanici_adi
```

İzin\_türü olarak :

ALL → tüm izinler, Insert → ekleme izni ve Update → değiştirme izni vb. izinler verilebilir.

Şimdi de “bulentvardal” kullanıcısına tüm hakları verelim.

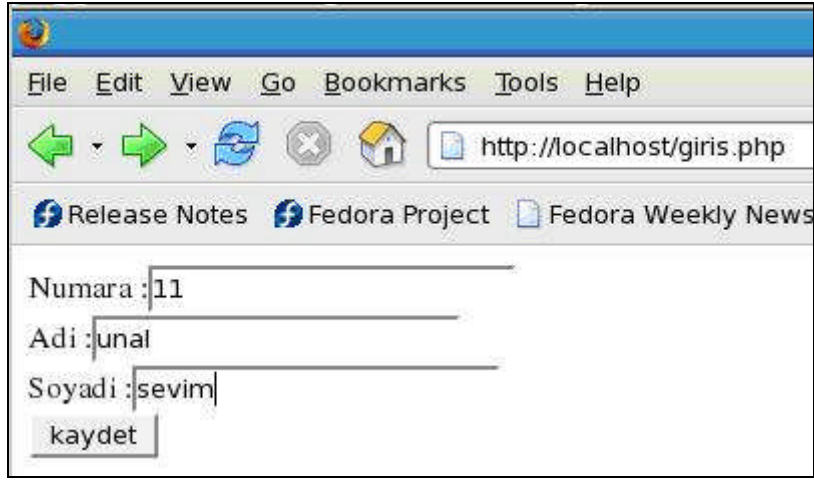
```
dosya=# GRANT ALL on ogrenciler TO bulentvardal;  
GRANT
```

Bu işlemler yapıldıktan sonra veri tabanına bağlantımızı PHP üzerinden yapabiliriz. Şimdi aşağıdaki kodları yazalım. Bir form içindeki text kutularına verileri gireceğiz.

```
vi /var/www/html/giris.php
```

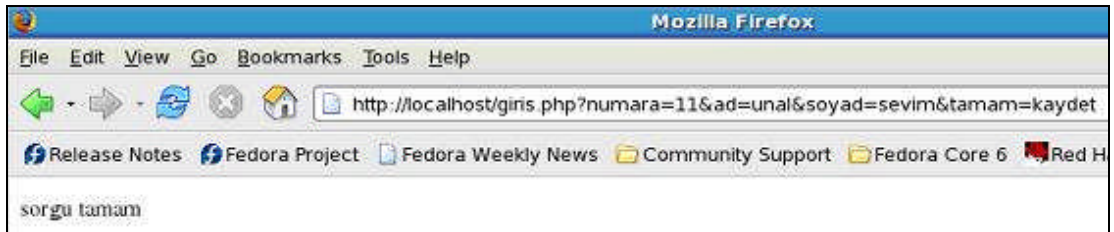
```
<HTML>  
<BODY>  
<?  
$ad=$_GET["ad"];  
$soyad=$_GET["soyad"];  
$numara=$_GET["numara"];  
if($ad==""){  
?>  
<form>  
Numara :<input type=text name=numara><br>  
Adi :<input type=text name=ad><br>  
Soyadi :<input type=text name=soyad><br>  
<input type=submit name=tamam value=kaydet>  
</form>  
<?  
}else{  
    $bag=pg_connect("dbname=dosya user=bulentvardal");  
    if(!$bag){  
        echo "Baglanti kurulamadi";  
    }else{  
        $sorgu=pg_query($bag,"insert into ogrenciler  
values($numara,'$ad','$soyad')");  
        echo " sorgu tamam";  
    }        }  
?>  
</BODY>  
</HTML>
```

Şimdi kodumuzu çalıştıralım.



Şekil 2.2: giris.php dosyasının Explorer penceresindeki ön izlemesi

Kaydet düğmesine basalım.



Şekil 2.3: giris.php dosyasındaki kaydet butonuna basıldıktan sonraki ön izlemesi

## 2.2. Veri Listeleme

Eğer php kodları ile bu dosya içindeki tüm kayıtları şu şekilde listelenir.

```
vi /var/www/html/listele.php
```

```
<HTML>
```

```
<BODY>
```

```
<?
```

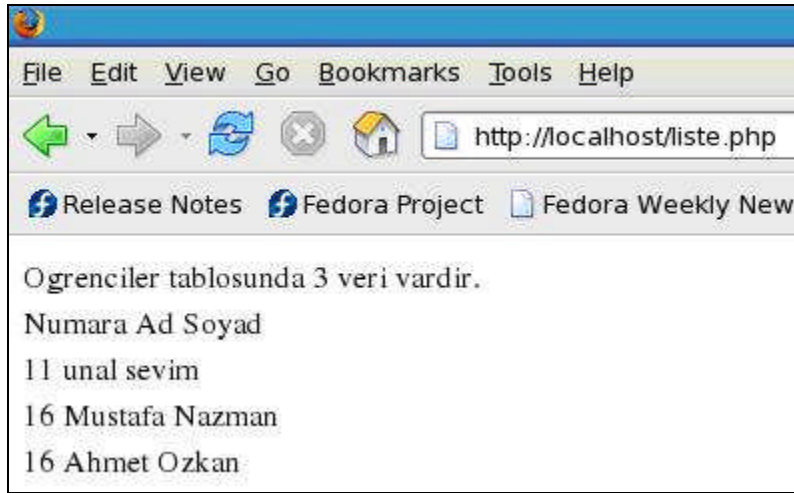
```
$bag=pg_connect("dbname=dosya user=bulentvardal");  
if(!$bag){  
    echo "Baglanti kurulamadi";  
}else{  
    $sorgu=pg_query($bag,"select * from ogrenciler");
```

```

$adet=pg_num_rows($sorgu);
echo "Ogrenciler tablosunda $adet veri vardır.";
echo "<br> Numara      Ad      Soyad<br>";
for($i=0;$i<=$adet;$i++){
    $numara=pg_fetch_result($sorgu,$i,0);
    $ad=pg_fetch_result($sorgu,$i,1);
    $soyad=pg_fetch_result($sorgu,$i,2);
    echo "$numara      $ad      $soyad<br>";
}
?>
</BODY>
</HTML>

```

Ve çalıştıralım.



**Şekil 2.4. liste.php dosyasının Explorer penceresindeki ön izlemesi**

Koddaki “pg\_num\_rows” ifadesi sorgu sonunda kaç satırlık veri elde edildiğini gösteriyor.

Yukarıdaki örnekte de görüldüğü üzere, önceden oluşturulan tablo içindeki isimleri listelemek için SELECT komutu kullanılıp almak istenilen sütun adı girilir. FROM komutundan sonra da tablonun adı girilip veri tabanındaki veriler listeleniyor. Tablodaki tüm sütunların verilerin listelenmesi istendiğinde (\*) parametresi kullanılır.

Şimdi de php kodları ile dosya içindeki ismi Ahmet olan kayıtları sorgulayalım.

```
vi /var/www/html/sorgu.php
```

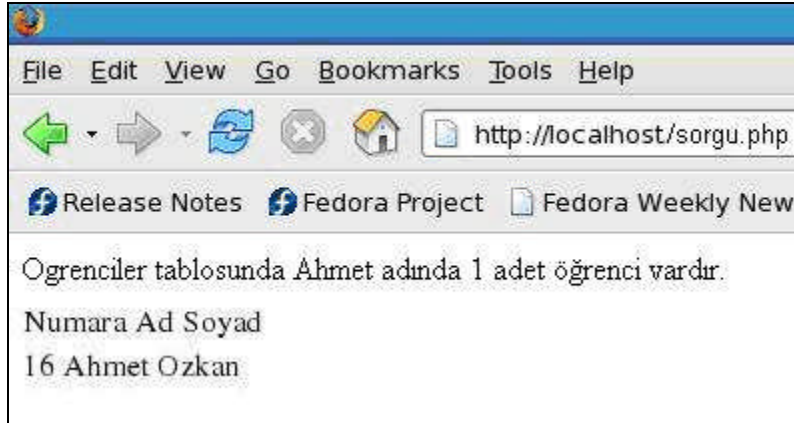
```
<HTML>
```

```

<BODY>
<?
    $bag=pg_connect("dbname=dosya user=bulentvardal");
    if(!$bag){
        echo "Baglanti kurulamadi";
    }else{
        $sorgu=pg_query($bag,"select * from ogrenciler
where ad = 'Ahmet'");
        $adet=pg_num_rows($sorgu);
        echo " Oğrenciler tablosunda Ahmet adında $adet öğrenci vardır.";
        echo "<br> Numara          Ad          Soyad<br>";
        for($i=0;$i<=$adet;$i++){
            $numara=pg_fetch_result($sorgu,$i,0);
            $ad=pg_fetch_result($sorgu,$i,1);
            $soyad=pg_fetch_result($sorgu,$i,2);
            echo "$numara          $ad          $soyad<br>";
        }
    }
?>
</BODY>
</HTML>

```

Ve çalıştıralım.



Şekil 2.5: sorgu.php dosyasının Explorer penceresindeki ön izlemesi

### 2.2.1. Alfabetik Olarak Veri Listeleme

SQL'den verileri listelenmesi istendiğinde veriler, SQL'de giriş sıralamasına göre listelenir. Belirli bir sütundaki verilerin, alfabetik sıralama ile listelenmesi istendiğinde ORDER BY veya GROUP BY komutları kullanılabilir.

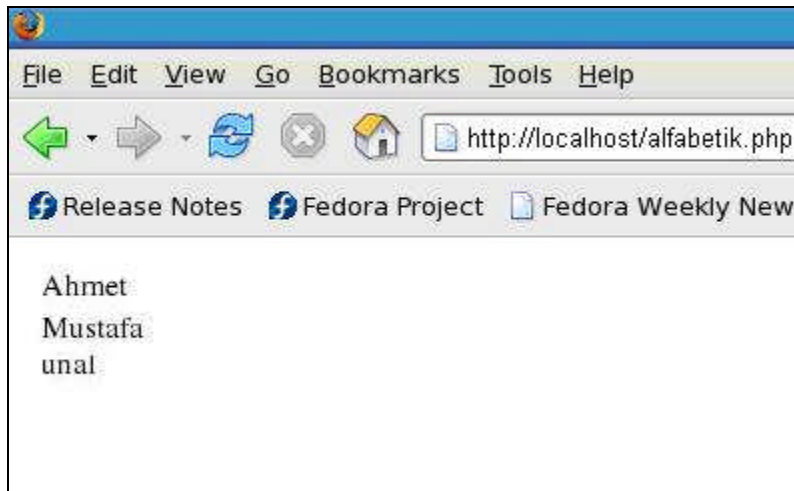
Önceki modüllerden de hatırlanacağı gibi tablo adından sonra ORDER BY komutu ve alfabetik veya rakamsal olarak verilerin sıralanması istenen sütun adı girilir.

Şimdi php kodları ile dosya içindeki isimleri alfabetik olarak listeleyelim.

```
vi /var/www/html/alfabetik.php
```

```
<HTML>
<BODY>
<?
    $bag=pg_connect("dbname=dosya user=bulentvardal");
    if(!$bag){
        echo "Baglanti kurulamadi";
    }else{
        $sorgu=pg_query($bag,"select ad from ogrenciler order by ad");
        for($i=0;$i<=$adet;$i++){
            $ad=pg_fetch_result($sorgu,$i,1);
            echo "$ad <br>";
        }
    }
?>
</BODY>
</HTML>
```

Ve çalıştıralım.



Şekil 2.6: alfabetik.php dosyasının explorer penceresindeki ön izlemesi



## 2.2.2. Verileri Tersten Listeleme

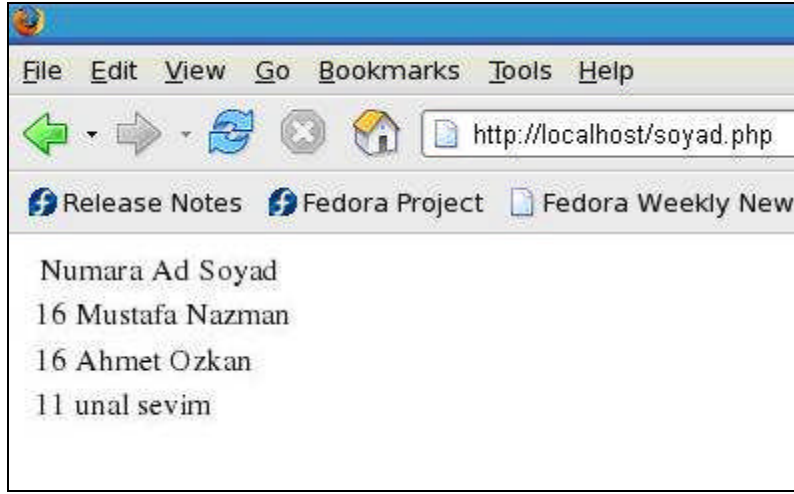
Verilerin alfabetik veya sayısal olarak tersi bir şekilde sıralanması istendiğinde DESC komutundan yararlanılır. Bu komut, tersine sıralaması yapılacak sütun adını girdikten sonra girilir.

Şimdi php kodları ile dosya içindeki kayıtları soyadlarının alfabetik olarak tersten sırasına göre listeleyelim.

```
vi /var/www/html/soyad.php
```

```
<HTML>
<BODY>
<?
    $bag=pg_connect("dbname=dosya user=bulentvardal");
    if(!$bag){
        echo "Baglanti kurulamadi";
    }else{
        $sorgu=pg_query($bag,"select * from ogrenciler order by soyad desc ");
        echo "<br> Numara      Ad      Soyad<br>";
        for($i=0;$i<=$adet;$i++){
            $numara=pg_fetch_result($sorgu,$i,0);
            $ad=pg_fetch_result($sorgu,$i,1);
            $soyad=pg_fetch_result($sorgu,$i,2);
            echo "$numara      $ad      $soyad<br>";
        }
    }
?>
</BODY>
</HTML>
```

Ve çalıştıralım.



Şekil 2.7: soyad.php dosyasının Explorer penceresindeki ön izlemesi

## 2.3. Veri Arama

Veri tabanının mantığı, mevcut birtakım verilere, en uygun tabloyu oluşturup tablolar içine verileri yükleyerek, gerektiğinde belirtilen ölçütlere göre listelemek olarak gösterilir. SQL sadece belirli isim ve belirli sayıların kaydedildiği sistem değildir. İstenilen bir metin hatta resim bile SQL'e depolanabilir.

1000 kelimelik bir metnin SQL'de kayıtlı olduğu bir veri tabanı düşününüz. Bu veri içinde istenilen bir kelimenin olup olmadığını anlamak için şimdiye kadar anlatılanlar yetersiz kalır. Bu gibi metinler içinde arama yapabilmek için SQL çeşitli komutlar sunmaktadır. Arama işlemini daha iyi kavrayabilmek için basit bir örnek verelim. Bu doğrultuda yeni bir tablo oluşturalım.

```
CREATE TABLE metin_tablosu (  
  
    id serial,  
    konu varchar(50) NOT NULL default '',  
    metin text NOT NULL,  
  
    CONSTRAINT id PRIMARY KEY(id)           );
```

```
INSERT INTO metin_tablosu  
VALUES (1, 'Macromedia ColdFusion MX', Java teknolojisi ile hazırlandı. Klasik C diliyle yazılan programda hafıza açığının olması, JAVA'nın çöp toplayıcısı özelliği özelliğini içinde barındırmadı, ColdFusion MX'in Java ile yazılmasında büyük bir etken olduğunu düşünüyorum.');
```

```
INSERT INTO metin_tablosu
```

*VALUES (2, 'JavaScript', 'HTML\'nin statik yapısına dinamik bir içerik eklemek amacıyla, 90\'lu yıllarda JavaScript dili geliştirilerek WEB\'de büyük bir devrim uyandırdı. Aradan yıllar geçmesi, büyük şirketlerin kodu, JavaScript\'in geliştirici firmasının tarayıcılarında çalışmıyor. Ne kadar ilginç... ');*

*INSERT INTO metin\_tablosu*

*VALUES (3, 'JavaAplet', 'Internet teknolojileri geliştikçe, web sayfalarını ziyaret eden kullanıcılara anlık değişimleri izletme amacıyla, JavaAplet denen bir olgu ortaya çıktı. Bu da Internet teknolojilerinde bir devrim yarattı. Zira JavaAplet\'ler yardımı ile artık, Chat ve Borsa endeksinin sayfayı güncelleme durumu olmadan anlık değişimleri görme olanağımız var.');*

Yukarıdaki sql kodları yazdıktan sonra veri tabanında aşağıdaki tablo oluşur.

id	Konu	Metin
1	Macromedia ColdFusion MX	<i>Java teknolojisi ile hazırlandı. Klasik C diliyle yazılan programda hafıza açığının olması, JAVA\'nın çöp toplayıcısı özelliği özelliğini içinde barındırmadı, ColdFusion MX\'in Java ile yazılmasında büyük bir etken olduğunu düşünüyorum.</i>
2	JavaScript	<i>'HTML\'nin statik yapısına dinamik bir içerik eklemek amacıyla, 90\'lu yıllarda JavaScript dili geliştirilerek WEB\'de büyük bir devrim uyandırdı. Aradan yıllar geçmesi, büyük şirketlerin kodu, JavaScript\'in geliştirici firmasının tarayıcılarında çalışmıyor. Ne kadar ilginç...</i>
3	JavaAplet	<i>Internet teknolojileri geliştikçe, web sayfalarını ziyaret eden kullanıcılara anlık değişimleri izletme amacıyla, JavaAplet denen bir olgu ortaya çıktı. Bu da Internet teknolojilerinde bir devrim yarattı. Zira JavaAplet\'ler yardımı ile artık, Chat ve Borsa endeksinin sayfayı güncelleme durumu olmadan anlık değişimleri görme olanağımız var.</i>

SQL'e veri girişi sırasında ayırt edici karakter olan tek tırnak kullanılıyor. Ama veri içinde tek tırnak kullanmak istendiğinde tek tırnağın ayırt edici karakter olmadığını belirtmek için ters bölü (\) işareti kullanılır. Genel olarak SQL'de arama fonksiyonu yerine getirebilmesi için LIKE kalıbı kullanılır. Yalnız bu komut kendi başına bir anlam ifade etmez. SQL'de arama yapmak isteniyorsa hangi sütün içinde arama yapılacağını belirtmek için WHERE kalıbının kullanılması gerek.

Şimdi php kodları ile dosyada içinde Aplet kelimesi geçen metinleri, konuları ile birlikte listeleyelim.

```
vi /var/www/html/aplet.php
```

```
<HTML>
```

```
<BODY>
```

```
<?
```

```

$bag=pg_connect("dbname=dosya user=bulentvardal");
if(!$bag){
    echo "Baglanti kurulamadi";
}
else{
$ sorgu=pg_query($bag,"select * from metin_tablosu where metin like '%Aplet%'");
$adet=pg_num_rows($sorgu);
echo "<table width='400' border='1' cellpadding='0' cellspacing='0' >"
echo "<tr> <td colspan=3 > Tabloda $adet kayitta Aplet kelimesi geçmektedir.“
echo "</td> </tr>";
echo "<tr><td > ID </td> <td > Konu </td> <td > Metin </td> </tr>";
    for($i=0;$i<=$adet;$i++){
        $id=pg_fetch_result($sorgu,$i,0);
        $konu=pg_fetch_result($sorgu,$i,1);
        $metin=pg_fetch_result($sorgu,$i,2);
echo "<tr><td > $id </td> <td > $konu </td> <td > $metin </td> </tr>";
    }
echo "</table>"
}
?>
</BODY>
</HTML>

```

Ve çalıştırılım.

Arama fonksiyonlarında kullanılabilir üç yöntem vardır.

*WHERE metin LIKE 'oze';* (I)

*WHERE metin LIKE 'ze%';* (II)

*WHERE metin LIKE '%ze%';* (III)

Yukarıda verilen birinci komut satırı, metin sütunu içinde verilerin sonu “ze” ile biten tüm verileri, ikinci komutta içinde “ze” olan başlayan tüm verileri, son komutta ise içinde “ze” hecesi geçen tüm verileri listeleyecektir.

## UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarını gerçekleştiriniz.

İşlem Basamakları	Öneriler
➤ Veri tabanında “kayit_no, ad, soyad, tel_no, adres” alanlarından oluşan bir tablo oluşturunuz.	➤ "Create table" komutundan faydalanınız.
➤ Php kodları ile, oluşturulan tabloya veri girişi yapmak için “kayit.php” dosyasını oluşturunuz.	
➤ Tüm kayıtları listeleyen “liste.php” dosyasını oluşturunuz.	
➤ Kayıtları soyadına göre tersten sıralayan “soyad.php” dosyasını oluşturunuz.	➤ Desc kalıbını sorgu cümlesinde kullanmayı unutmayınız.
➤ Adres alanında “Konya“ kelimesi geçen kayıtları listeleyen “adres.php” dosyasını oluşturunuz.	➤ Sorgu cümlesinde % Konya % biçimini kullanmayı unutmayınız.

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerde verilen bilgiler doğru ise (D), yanlış ise (Y) yazınız.

1. Alandaki verileri tersten sıralamak için “order by” komutu tek başına yeterlidir. (D) (Y)
2. “lar” ile biten kelimelerin bulunduğu kayıtları listelemek için ‘%lar’ ifadesi kullanılır. (D) (Y)
3. Veri arama işlemleri için “aplet” komutu kullanılır.(D) (Y)

**Aşağıdaki “php” komut satırlarındaki hataları düzeltiniz.**

4. `for(i=0;i<=adet;i++)`
5. `echo (<br> Ad Soyad<br>);`

### 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-3

## AMAÇ

Betik dili (PHP) ile veri tabanında veri güncelleme ve silme işlemlerini sorunsuz bir şekilde gerçekleştirebileceksiniz.

## ARAŞTIRMA

- İnternet ve çeşitli kaynaklardan PHP dili ile veri tabanı kontrol eden programları inceleyiniz.

## 3. PHP İLE VERİ GÜNCELLEME VE SİLME

### 3.1. Veri Güncelleme

Veri tabanındaki tablolara girilen verilerin, bazı durumda güncellenmesi istenebilir. Örneğin bir firma için müşteri kaydı yapılıyor. Müşterinin telefon numarasını değiştirmek istendiğinde aşağıdaki SQL cümlecisi kullanılabilir.

```
UPDATE ilk_tablo SET 03325554433 WHERE ad "Ahmet BAYRAM";
```

Şimdi bir önceki öğrenme faaliyetinde oluşturulan veri tabanındaki öğrenciler tablosunda istenen veriyi istenen değerle değiştiren php dosyasını oluşturalım.

```
vi /var/www/html/gucel.php
```

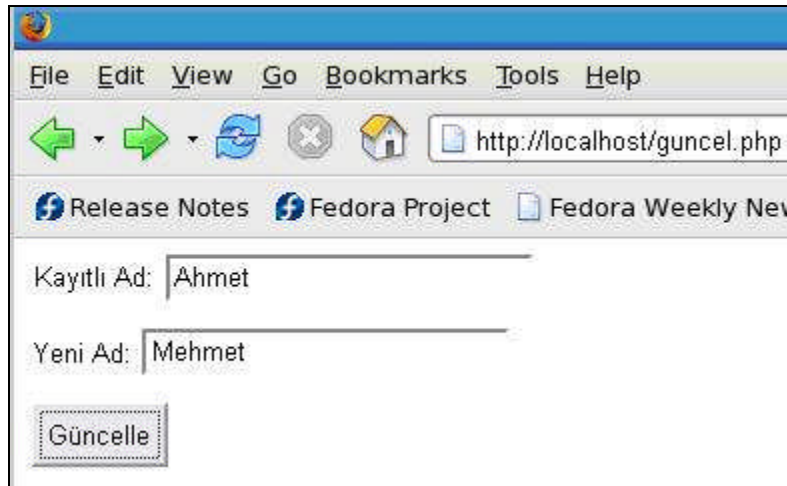
```
<HTML>
<BODY>
<?
$ad=$_GET["ad"];
$yeniad=$_GET["y_ad"]
if($ad==""){
?>
<form>
Kayıtlı Ad :<input type=text name="ad"><br>
Yeni Ad :<input type=text name="y_ad"><br>
<input type=submit name="tamam" value="Guncelle">
</form>
<?
}
else
```

```

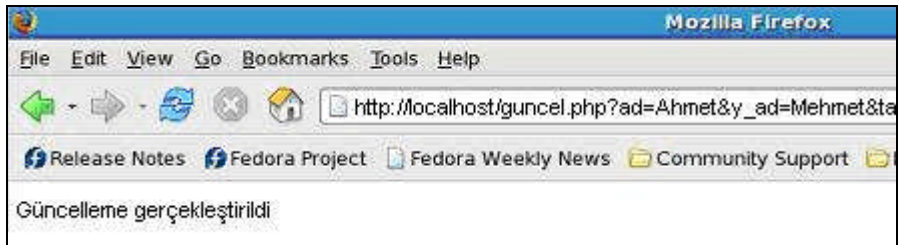
{
$bag=pg_connect("dbname=dosya user=bulentvardal");
if(!$bag){
    echo "Baglanti kurulamadi";
}
else
{
$sorgu=pg_query($bag," update ogrenciler set ad=$y_ad where ad=$ad");
    echo "Güncelleme gerçekleştirildi";
}
}
?>
</BODY>
</HTML>

```

Şimdi kodumuzu çalıştıralım.



Şekil 3.1: guncel.php dosyasının Explorer penceresindeki ön izlemesi



Şekil 3.2: guncel.php dosyasındaki “Güncelle” butonuna basıldıktan sonraki ön izlemesi



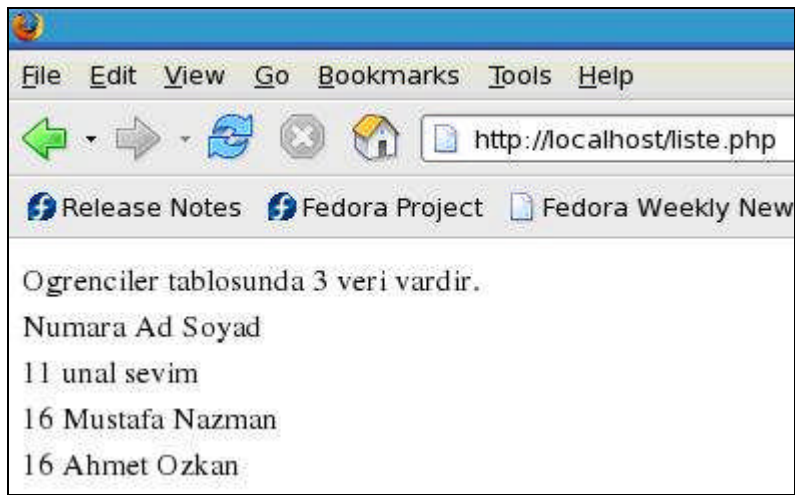
## 3.2. Veri Sime

Şimdi de silme programı yapalım.

```
vi /var/www/html/silme.php
```

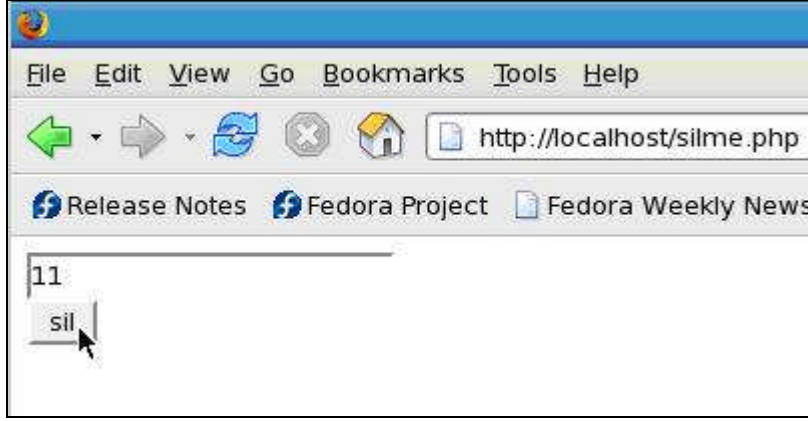
```
<HTML>
<BODY>
<?
$silnum=$_GET["silnum"];
if($silnum==""){
    echo "<form>";
    echo "<input type=text name=silnum>";
    echo "<br><input type=submit name=tamam value=sil>";
    echo "</form>"; }
else{
    $bag=pg_connect("dbname=dosya user=bulentvardal");
    if(!$bag){
        echo "Baglanti kurulamadi";
    }else {
        $sorgu=pg_query($bag,"delete from ogrenciler where numara=$silnum");
    } }
?>
</BODY>
</HTML>
```

Şimdi de ekran çıktısına bakalım. Örneğin 11 nu.lı öğrenciyi silelim. Silmeden önce listeleme yapalım.



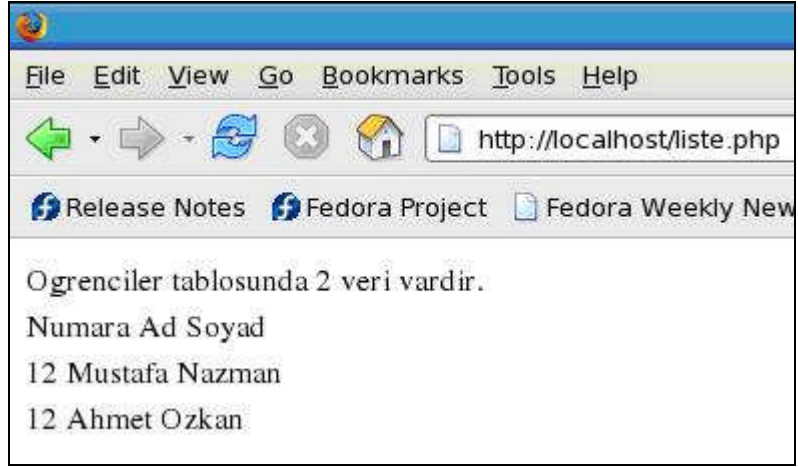
Şekil 3.3: liste.php dosyasının silme.php dosyası çalıştırılmadan önceki ön izlemesi

Sonra “silme.php” dosyasını çağırıp 11 değerini girdikten sonra sil butonuna basalım.



**Şekil 3.3: Veri tabanından 11 numaralı öğrencinin silinmesi**

Şimdi tekrar “liste.php”yi çağıralım.



**Şekil 3.3: 11 numaralı öğrenci silindikten sonraki durum**

## UYGULAMA FAALİYETİ

Aşağıdaki işlem basamaklarını gerçekleştiriniz.

İşlem Basamakları	Öneriler
<p>➤ Bir önceki öğrenme faaliyetinde oluşturduğunuz örnek php uygulamasında adı Alparslan olan kayıtları Alpaslan olarak güncelleyen php dosyasını oluşturunuz.</p>	<p>➤ Bir önceki öğrenme faaliyetindeki php uygulamasını kontrol ettikten sonra bu uygulama faaliyetine başlayınız.</p>
<p>➤ Rehberde kayıtlı olan Sevgi Akman Konya'dan Ankara'ya taşınmıştır. Adresi güncelleyen php dosyasını oluşturunuz.</p>	
<p>➤ Kayıt numarasına göre silme yapan php dosyasını oluşturunuz.</p>	
<p>➤ Rehberde kayıtlı olan Ayşe Aydın, Mehmet Yılmaz ile evlenmiştir. Gerekli güncellemeyi yapan php dosyasını oluşturunuz.</p>	
<p>➤ Rehberde kayıtlı olan Erkan Server GSM operatörünü değiştirmiştir. Gerekli güncellemeyi yapan php dosyasını oluşturunuz.</p>	

## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümlelerde verilen bilgiler doğru ise (D), yanlış ise (Y) yazınız.

1. Tablodaki alandan kayıt silmek için del komutu kullanılır.(D) (Y)
2. “update” komutu kayıttaki bilgileri değiştirmek için kullanılır. (D) (Y)

Aşağıdaki php komut satırlarındaki hataları belirtiniz.

3. update from rehber set ad=y\_ad where \$ad=ad
4. del ogrenciler where numara=silnum
5. if(!\$bag) then echo "Baglanti kurulamadi";

### 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

## KONTROL LİSTESİ

Bu modül kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri **Evet** ve **Hayır** kutucuklarına ( X ) işareti koyarak kontrol ediniz.

Değerlendirme Ölçütleri				
		Evet	Hayır	
1	Faaliyetler için gerekli arařtırmaları eksiksiz yaptınız mı?			
2	Faaliyet için gereken araç gereci ve dokümanı temin ettiniz mi?			
3	İstenilen haklara sahip PostgreSQL kullanıcıları oluşturduunuz mu?			
4	PHP scripti yazabildiniz mi?			
5	PHP ile PostgreSQL arasında bağlantı kurdunuz mu?			
6	PHP ile PostgreSQL arasında bağlantı çeşitlerini doğru kullanıp kurabildiniz mi?			
7	İstenilen özellikte alan ve tablolardan oluşan veri tabanı oluşturduunuz mu?			
8	Oluşturulan alanlara PHP dilinde kayıtlar girdiniz mi?			
9	Veri tabanındaki kayıtları web tarayıcıda listelediniz mi?			
10	İstenilen alanlar üstünde güncelleme işlemini yaptınız mı?			
11	İstenilen alanlar üstünde silme işlemini yaptınız mı?			
12	Sayısal verileri büyükten küçüğe sıraladınız mı?			
13	Sayısal verileri küçükten büyüğe sıraladınız mı?			
14	Alfasayısal SQL sonuçlarını alfabetik olarak sıralayabildiniz mi?			
15	Alfasayısal SQL sonuçlarını alfabetik olarak tersten sıralayabildiniz mi?			
16	Metin türündeki verilerde istenilen özelliklerde kelime araması yaptırabildiniz mi?			
17	Kayıt güncelleme işlemlerini doğru ve eksiksiz yapabildiniz mi?			
18	Kayıt silme işlemlerini doğru ve eksiksiz yapabildiniz mi?			
19	Karşılaşılan hataları düzeltebildiniz mi?			
20	Düzeltilemediğiniz hataları arkadaşlarınızla yardımlaşarak düzeltebildiniz mi?			
<b>TOPLAM PUAN</b>				

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ -1'İNCEVAP ANAHTARI

1	Doğru
2	Doğru
3	Yanlış
4	C
5	B

## ÖĞRENME FAALİYETİ -2'NİN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	Yanlış
4	<code>for(\$i=0;\$i&lt;=\$adet;\$i++)</code>
5	<code>echo "&lt;br&gt; Ad Soyad&lt;br&gt;";</code>

## ÖĞRENME FAALİYETİ -3'ÜN CEVAP ANAHTARI

1	Yanlış
2	Doğru
3	<code>update rehber set ad=\$y_ad where ad=\$ad</code>
4	<code>del from ogrenciler where numara=\$silnum</code>
5	<code>if(!\$bag) echo "Baglanti kurulamadi";</code>

Cevaplarınızı cevap anahtarları ile karşılaştırarak kendinizi değerlendiriniz.

## KAYNAKÇA

- MASUDA Youchi, Bülent VARDAL, **Web Sistem Uygulamaları**, MEB Yayınları, 2005.
- ŞAMLI Mehmet, **MySQL ve PostgreSQL İle Veritabanı Programlama**, Pusula Yayıncılık, İstanbul, 2003.