

FreeBSD üzerinde PostgreSQL Temel Kurulum ve Yapılandırması

Bu makale de FreeBSD işletim sistemi üzerine PostgreSQL 8.0 (8.0.7) kurulumu ve yapılandırması anlatılacaktır.

Bu belgenin en son haline

http://www.enderunix.org/docs/postgresql/freebsd_postgresql.pdf adresinden ulaşabilirsiniz.

Yazar: İsmail YENİGÜL

EnderUNIX Çekirdek Takım Üyesi

ismail at enderunix dot org

ismail.yenigul at endersys dot com

<http://www.enderunix.org>

21 Mar 2006 Sal EET 13:52:35 – İlk Yazım

29 Mar 2006 Çar EEST 18:02:52 – Bir kac örnek eklendi.

29 Mar 2006 Çar EEST 23:20:21 – PostgreSQL çalıştırma kısmında düzeltmeler yapıldı.

İçindekiler

1.	PostgreSQL Kurulumu.....	2
2.	Yapılandırma.....	2
2.1.	PostgreSQL’i Çalıştırma.....	2
2.2.	Veritabanı Oluşturma.....	4
2.2.1.	createdb Komutu İle.....	4
2.2.2.	SQL Komutu İle.....	5
2.3.	Veritabanlarını Listeleme.....	5
2.4.	Veritabanı Kullanıcısı Ekleme.....	6
2.4.1.	Komut Satırından.....	6
2.4.2.	SQL Komutu ile.....	6
2.5.	Kullanıcı Silme.....	8
2.5.1.	Komut Satırından.....	8
2.5.2.	SQL ile.....	8
2.6.	Kullanıcı Parolası Değişirme.....	8
2.7.	Kullanıcı Grubu Oluşturma.....	9
2.7.1.	Gruba Yeni Üye Ekleme.....	9
2.7.2.	Grubtan Üye Çıkarma.....	9
2.7.3.	Grupları Listeleme.....	9
2.8.	Tablo İşlemleri.....	10
2.8.1.	Tablo Oluşturma.....	10
2.8.2.	Kayıt Ekleme.....	10
2.8.3.	Kayıt Listeleme.....	10
3.	Yedekleme.....	11
3.1.	Yedek Alma.....	11
3.2.	Yedekten Geri Dönme.....	11
4.	pg_ctl.....	12

5. PHP'ye PostgreSQL Desteđi	12
6. phpPgAdmin Kurulumu.....	13
7. Kaynaklar:.....	14

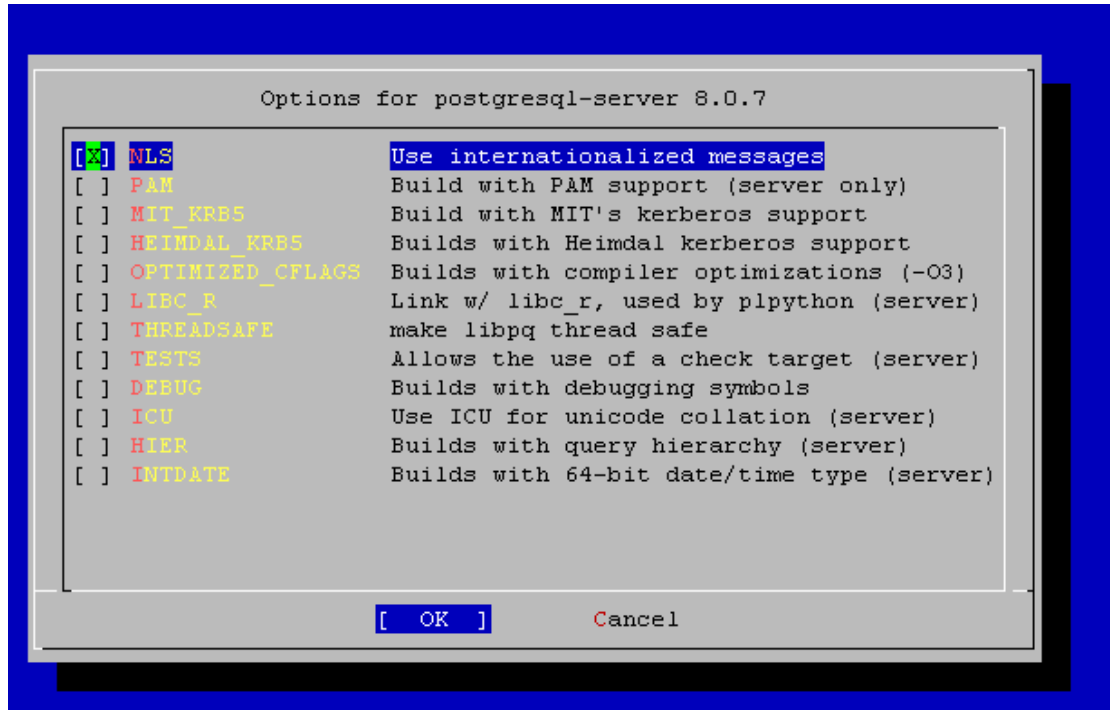
1. PostgreSQL Kurulumu

PostgreSQL kurulumu port ağacından yapılacaktır. Bu yüzden kurulum yapmadan önce port ağacınızı güncellemenizde fayda var.

```
# cd /usr/ports/databases/postgresql80-server/
```

```
# make install clean
```

Bu komut verildikten sonra aşağıdaki gibi bir ekran gelecektir. Ön tanımlı değerlerle kurmak için OK tuşuna basılır.



2. Yapılandırma

2.1. PostgreSQL'i Çalıştırma

PostgreSQL'in çalışması ve açılıştta başlaması için /etc/rc.conf dosyasına aşağıdaki satır yazılır

```
postgresql_enable="YES"
```

İk veritabanını oluşturmak için rc.conf dosyasına yukarıdaki ifadeyi yazdıktan sonra root olarak aşağıdaki komut verilir.

Not: Eđer PostgreSQL'de Tőrķçe desteęini etkinleřtirmek isterseniz ařaęıdaki komutu vermeden önce /etc/profile dosyasına ařaęıdaki iki satır eklenir.

```
export LANG=tr_TR.ISO8859-9
export MM_CHARSET=ISO-8859-9
```

Daha sonra bu deęiřkenlerin etkinleřmesi iin sistemden ıkılıp tekrar girilir.

Bu evre deęiřkenlerin tanımlı olup olmadıęını anlamak iin ařaęıdaki komutlar verilebilir.

```
# echo $LANG
tr_TR.ISO8859-9
# echo $MM_CHARSET
ISO-8859-9
```

Artık PostgreSQL'in bařlangı deęerlerini oluřturmak iin (initialize) ařaęıdaki komut verilir.

```
# /usr/local/etc/rc.d/010.pgsql.sh initdb
```

Bu veritabanı sistemine ait olan dosyaların sahibi "pgsql" kullanıycısı olacaktır.

Bu kullanıcı aynı zamanda sunucu sőrrecinin de sahibi olmalıdır.

Veritabanı kumesi tr_TR.ISO8859-9 yereli ile ilklendirilecek. Öntanımlı veritabanı dil kodlaması LATIN5 olarak ayarlandı.

```
/usr/local/pgsql/data dizini yaratılıyor... tamam
/usr/local/pgsql/data/global dizini yaratılıyor... tamam
/usr/local/pgsql/data/pg_xlog dizini yaratılıyor... tamam
/usr/local/pgsql/data/pg_xlog/archive_status dizini yaratılıyor...
tamam
/usr/local/pgsql/data/pg_clog dizini yaratılıyor... tamam
/usr/local/pgsql/data/pg_subtrans dizini yaratılıyor... tamam
/usr/local/pgsql/data/base dizini yaratılıyor... tamam
/usr/local/pgsql/data/base/1 dizini yaratılıyor... tamam
/usr/local/pgsql/data/pg_tblspc dizini yaratılıyor... tamam
ön tanımlı max_connections seçiliyor ... 40
ön tanımlı shared_buffers seçiliyor ... 1000
yapılandırma dosyaları yaratılıyor ... tamam
/usr/local/pgsql/data/base/1 içinde template1 veritabanı
yaratılıyor.tamam
pg_shadow hazırlanıyor ... tamam
Sistem tabloları için sınırsız satır sayısı etkinleştiriliyor ...
tamam
pg_depend hazırlanıyor ... tamam
sistem viewları yaratılıyor ... tamam
pg_description yükleniyor ... tamam
dönümler yükleniyor ... tamam
gömülü nesnelerdeki izinler ayarlanıyor ... tamam
information schema yaratılıyor ... tamam
template1 veritabanı vakumlanıyor ... tamam
template1 template0'a kopyalanıyor ... tamam
```

UYARI: Yerel bađlantılar için "trust" yetkilendirmesi etkinleştiriliyor.

Bunu, pg_hba.conf dosyasını düzenleyerek ya da initdb'yi yeniden alıptırdığınızda -A parametresi ile deđiptirebilirsiniz..

Ýplem bađarılı. Veritabanı sunucusunu:

```
/usr/local/bin//postmaster -D /usr/local/pgsql/data
```

```
ile ya da
/usr/local/bin//pg_ctl -D /usr/local/pgsql/data -l logfile start
ile başlatabilirsiniz.
```


Yukarıdaki çıktılarda görüldüğü gibi PostgreSQL veritabanı dizini olarak /usr/local/pgsql/data/ kullanılmıştır. Bu dizini PostgreSQL dünyasında bu dizine “Database Cluster” adı verilir. initdb parametresinin verilmesi ile template1 ve template0 olmak üzere iki adet veritabanı oluşturulur. template1 veritabanı bir kalıp (template) veritabanıdır. Bu veritabanı üzerinde oluşturulacak her türlü nesnelere (objects) bundan sonraki oluşturulacak yeni veritabanlarında da oluşturulacaktır.

PostgreSQL’i başlatmak için ise aşağıdaki komut verilir.

```
# /usr/local/etc/rc.d/010.pgsql.sh start
```

PostgreSQL’in çalışıp çalışmadığına ps komutu ile bakılabilir.

```
# ps auxw | grep pgsq
pgsql      75606  0,0  1,2  6900 3120  ??  I    11:18PM  0:00,00
postmaster: stats collector process (postgres)
pgsql      75605  0,0  1,2  7832 3060  ??  S    11:18PM  0:00,00
postmaster: stats buffer process (postgres)
pgsql      75604  0,0  1,3 16148 3328  ??  S    11:18PM  0:00,00
postmaster: writer process (postgres)
pgsql      75602  0,0  1,5 16124 3820  ??  Is   11:18PM  0:00,35
/usr/local/bin/postmaster -D /usr/local/pgsql/data (postgres)
root       75614  0,0  0,2  1092  584  p0  R+   11:19PM  0:00,01 grep
pgsql
#
```

ps çıktısında da görüldüğü gibi PostgreSQL **pgsql** kullanıcısı haklarıyla çalışmaktadır. Tüm veritabanı yönetim işlemlerinde de bu kullanıcı kullanılmaktadır. Bu kullanıcı her işletim sisteminde farklı olabilir. PostgreSQL dokümanlarında kurulumlarında kullanıcı adı olarak **postgres** kabul edilmiştir.

```
# su - pgsq
$ pg_ctl status
pg_ctl: postmaster çalışıyor (PID: 75602)
/usr/local/bin/postgres -D /usr/local/pgsql/data
$
```

2.2. Veritabanı Oluşturma

Veritabanı iki türlü oluşturulabilir:

2.2.1. createdb Komutu İle

```
# su - pgsq
$ createdb deneme1
CREATE DATABASE
```

\$

Türkçe karakter setli veritabanı oluşturmak için:

```
$ createdb -E LATIN5 deneme1
CREATE DATABASE
$
```

2.2.2. SQL Komutu İle

```
# su - pgsq1
$ psql template1
```

```
template1=# CREATE DATABASE deneme2;
```

```
CREATE DATABASE
template1=#
```

Türkçe karakter setli veritabanı oluşturmak için:

```
template1=# CREATE DATABASE deneme2 WITH ENCODING 'LATIN5';
CREATE DATABASE
template1=#
```

2.3. Veritabanlarını Listeleme

Var olan veritabanlarını listelemek için ise aşağıdaki sorgu kullanılabilir.

```
# su - pgsq1
```

```
devel~$ psql -l
      List of databases
  Name | Owner | Encoding
-----+-----+-----
deneme | pgsq1 | SQL_ASCII
deneme2 | pgsq1 | SQL_ASCII
template0 | pgsq1 | SQL_ASCII
template1 | pgsq1 | SQL_ASCII
(4 rows)
```

Veya

```
$ psql template1
template1=# select datname from pg_database;
 datname
-----
 template1
 template0
 deneme
 deneme2
(4 rows)
template1=#
```

Veya

```

template1=# \l
      List of databases
  Name          | Owner   | Encoding
  -----+-----+-----
  deneme        | pgsq1   | SQL_ASCII
  deneme2       | pgsq1   | SQL_ASCII
  template0     | pgsq1   | SQL_ASCII
  template1     | pgsq1   | SQL_ASCII
  (4 rows)

template1=# \q
$

```

2.4. Veritabanı Kullanıcısı Ekleme

Öncelikle PostgreSQL'deki kullanıcı yapısı MySQL'den oldukça farklıdır. MySQL'de veritabanına göre kullanıcı tanımlanmaktaydı. PostgreSQL'de ise bir kullanıcı ön tanımlı olarak tüm veritabanlarına ulaşabilmekte ve bu veritabanları üzerinde tablo vs oluşturabilmektedir. Fakat bir kullanıcının oluşturduğu tabloya diğer kullanıcılar –Eğer sonradan özel bir izin verilmemişse – erişememektedir.

Not: PostgreSQL veritabanı kullanıcıları UNIX sistem kullanıcılarından farklı kullanıcılarıdır.

2. Türlü kullanıcı ekleme yöntemi vardır.

2.4.1. Komut Satırından

```

# su - pgsq1
$ createuser
Enter name of user to add: enderunix
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) y
CREATE USER
$ exit
#

```

Burada enderunix kullanıcına superuser (veritabanı ve başka kullanıcılar oluşturabilme) hakkı verilmiştir.

2.4.2. SQL Komutu ile

SQL ile herhangi bir veritabanına ulaşarak kullanıcı oluşturulabilir.

```

# su - pgsq1
$ psql deneme

```

Welcome to psql 8.0.7, 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
```

```
deneme=# CREATE USER acikakademi with password 'akademi123';
CREATE USER
deneme=#
```

Burada oluşturulan acikakademi kullanıcısı normal bir kullanıcısıdır. Bu kullanıcıya veritabanı oluşturma hakkı vermek için satır sonuna CREATEDB başka kullanıcılar oluşturma izni vermek için ise CREATEUSER parametresi eklenebilir.

Örnek:

```
# su - pgsql
$ psql template1
```

```
template1=# CREATE USER endersys with password 'pass0' CREATEUSER;
CREATE USER
template1=# CREATE USER acikkod with password 'pass1' CREATEUSER
CREATEDB;
CREATE USER
template1=#
```

Var olan kullanıcıları listelemek için ise aşağıdaki komut verilir.

```
deneme=# SELECT username from pg_user;
username
-----
pgsql
enderunix
acikakademi
endersys
acikkod
(5 rows)
deneme=#
```

Veya

```
template1=# \du
                                List of users
  User name | User ID |          Attributes          | Groups
-----+-----+-----+-----
acikakademi |      102 |                               |
acikkod     |      107 | superuser, create database |
endersys    |      106 | superuser                    |
enderunix   |      100 | superuser, create database |
pgsql       |         1 | superuser, create database |
(5 rows)
template1=#
```

Yeni oluşturulan kullanıcılar ile veritabanına bağlanmak için psql komutuna -U parametresi kullanılabilir. -d ile veritabanı adı belirtilebilir.

Eğer veritabanı kullanıcısı ile psql komutunu çalıştıran sistem kullanıcısının adı aynı ise -U parametresinin verilmesine gerek yoktur.

```
# su - endersys
$ psql deneme
Welcome to psql 8.0.7, 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
```

```
deneme=#
```

Veya

```
# su - pgsq1
$ psql -U endersys -d deneme
Welcome to psql 8.0.7, 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
```

```
deneme=#
```

2.5. Kullanıcı Silme

2.5.1. Komut Satırından

```
# su - pgsq1
$ dropuser ismail
DROP USER
$
```

2.5.2. SQL ile

```
# su - pgsq1
$ psql deneme

deneme=# drop USER ismail;
DROP USER
deneme=# \q
```

2.6. Kullanıcı Parolası Değişirme

```
# su - pgsq1
$ psql deneme
templatel=# ALTER USER ismail with password 'deneme123';
ALTER USER
templatel=#
```


2.7. Kullanıcı Grubu Oluşturma

UNIX'te olduğu gibi PostgreSQL'de de gruplar oluşturup kullanıcılar gruplara dahil edilebilir. Bu sayede bir gruba verilen izin gruptaki tüm kullanıcılar için geçerli olacaktır.

```
# su - pgsq1
psql template1
Welcome to psql 8.0.7, 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

template1=# CREATE GROUP yonetici WITH USER endersys;
CREATE GROUP
```

2.7.1. Gruba Yeni Üye Ekleme

```
template1=# ALTER GROUP yonetici ADD USER endersys;
ALTER GROUP
```

2.7.2. Grubtan Üye Çıkarma

```
template1=# ALTER GROUP yonetici DROP USER endersys;
ALTER GROUP
```

2.7.3. Grupları Listeleme

```
template1=# SELECT groname FROM pg_group;
 groname
-----
 yonetici
(1 row)
```

Veya

```
template1=# \dg
List of groups
Group name | Group ID
-----+-----
 yonetici  |      100
(1 row)
```

```
template1=# \du
List of users
User name | User ID | Attributes | Groups
-----+-----+-----+-----
 acikakademi |      102 |             |
 acikkod     |      107 | superuser, create database |
 endersys    |      106 | superuser   | {yonetici}
```

```
enderunix | 100 | superuser, create database | {yonetici}
pgsql | 1 | superuser, create database |
(5 rows)
```

```
template1=#
```

Oluşturulan gruplara aşağıdaki gibi tablolara erişim izni verilir.

```
deneme2=# GRANT SELECT,INSERT,UPDATE ON siparis TO GROUP yonetici;
GRANT
```

Burada deneme2 veritabanındaki siparis tablosu için yonetici grubuna SELECT, INSERT, UPDATE hakkı verilmiştir. Aynı anda hem grup hem de kullanıcıya aşağıdaki şekilde izin verilebilir.

```
deneme2=# GRANT SELECT, INSERT, UPDATE, TRIGGER ON "unix" TO
"ismail", GROUP "deneme" WITH GRANT OPTION
deneme2=#
```

2.8. Tablo İşlemleri

2.8.1. Tablo Oluşturma

```
deneme2=# create table test (id serial, fname varchar(20),lname
varchar(20));
NOTICE: CREATE TABLE will create implicit sequence "test_id_seq" for
serial column "test.id"
CREATE TABLE
deneme2=#
```

2.8.2. Kayıt Ekleme

```
deneme2=# INSERT INTO test(fname,lname) VALUES ('ali', 'veli');
INSERT 17273 1
deneme2=# INSERT INTO test(fname,lname) VALUES ('49', '50');
INSERT 17274 1
```

2.8.3. Kayıt Listeleme

```
deneme2=# select * from test;
 id | fname | lname
----+-----+-----
  1 | ali   | veli
  2 | 49    | 50
(2 rows)
```

3. Yedekleme

3.1. Yedek Alma

Yedekleme işlemi için `pg_dump` komutu kullanılabilir. Bu komut tek bir veritabanı yedeği alır.

```
# su - postgres
$ pg_dump deneme >deneme.sql
```

Tüm veritabanlarının yedeğini almak için `pg_dumpall` komutu kullanılabilir.

```
$ pg_dumpall >all.sql
```

Ek olarak PostgreSQL FreeBSD port ağacından kurulduğunda `/usr/local/etc/periodic/daily/502.pgsql` dosyası oluşur. Bu betik her gün yedek almak ve vacuum işlemi için kullanılır. Vacuum ve yedek işlemini etkinleştirmek için

`/etc/periodic.conf` dosyası yoksa oluşturulup içine aşağıdaki satırlar yazılır.

```
daily_postgresql_vacuum_enable="YES"
daily_postgresql_backup_enable="YES"
```

Bu işlemden sonra `~pgsql/backups/` altına tüm veritabanlarının yedeği alınacaktır.

```
# ls -l ~pgsql/backups/
total 14
drwx----- 2 postgres postgres 512 Mar 19 14:49 ./
drwxr-xr-x 4 postgres postgres 512 Mar 19 14:41 ../
-rw-r--r-- 1 postgres postgres 2790 Mar 19 14:49 pgdump_deneme2_2006-03-19T14:48:58
-rw-r--r-- 1 postgres postgres 1761 Mar 19 14:49 pgdump_deneme_2006-03-19T14:48:58
-rw-r--r-- 1 postgres postgres 1951 Mar 19 14:48 pgdump_template1_2006-03-19T14:48:58
-rw-r--r-- 1 postgres postgres 341 Mar 19 14:48 pgglobals_2006-03-19T14:48:58.gz
#
```

3.2. Yedekten Geri Dönme

Bir önceki örnekte yedeği alınan deneme veritabanının `deneme.sql` dosyasından yüklemek için aşağıdaki gibi bir komut verilir.

```
# psql -d deneme -f deneme.sql
pg_dumpall ile alınan tüm yedeği dönmek için ise aşağıdaki gibi bir komut verilir.
```

```
# psql -f all.sql template1
```

ikinci örnekte `pg_dumpall` çıktısından yedeği dönerken hangi veritabanına bağlanıldığının bir önemi yoktur. Bu çıktının içerisinde veri eklerken `\connect deneme` `\connect template1` şeklinde hangi veritabanına bağlanması gerektiğini belirten ibareler vardır.

4. pg_ctl

pg_ctl PostgreSQL'i başlatmak, durdurmak, yeniden başlatmak ve servisin durumu hakkında bilgi veren bir uygulamadır.

Kullanım şekilleri:

```
pg_ctl start [-w] [-s] [-D datadir] [-l filename] [-o options] [-p path]  
pg_ctl stop [-W] [-s] [-D datadir] [-m s[mart] | f[ast] | i[mmediate]]  
pg_ctl restart [-w] [-s] [-D datadir] [-m s[mart] | f[ast] | i[mmediate]] [-o options]  
pg_ctl reload [-s] [-D datadir]  
pg_ctl status [-D datadir]  
pg_ctl kill [signal_name] [process_id]
```

-D: data dizini

-l: Kayıt (log) dosyasının yolunu belirtir.

-m: servisin hangi kipte kapatılacağını belirtir. Üç farklı tip vardır.

- Smart: Tüm istemciler bağlantıyı koparana kadar bekler. (Ön tanımlı)
- Fast: İstemcilerin bağlantıyı koparmasını beklemez. Tüm aktif işlemler (transaction) geri alınır (roll back) ve servis kapatılır.
- Immediate: Tüm PostgreSQL prosesleri düzgün kapanmadan sonlandırılır. Bu işlem servisin tekrar açıldığında kurtarma (recovery) işleminin yapılmasına sebep olur.

Örnekler:

```
# su - pgsq1  
$ pg_ctl status  
pg_ctl: postmaster is running (PID: 17264)  
/usr/local/bin/postgres -D /usr/local/pgsq1/data  
$ pg_ctl start -D /usr/local/pgsq1/data -l logfile  
postmaster starting  
$ pg_ctl stop -D /usr/local/pgsq1/data -m smart  
waiting for postmaster to shut down.... done  
postmaster stopped
```

5. PHP'ye PostgreSQL Desteği

/usr/ports/Mk/bsd.database.mk dosyasında ön tanımlı PostgreSQL versiyonu 7.4'dür. Burada 8.0 serisini kullandığımız için DEFAULT_PGSQL_VER çevre değişkenini 80 olarak tanımlamamız gerekiyor.

```
# cd /usr/ports/databases/php4-pgsql  
# export DEFAULT_PGSQL_VER=80  
# make install clean
```

Kurulumdan sonra httpd.conf dosyasında aşağıdaki satırlar yoksa eklenir ve apache yeniden başlatılır.

```
AddType application/x-httpd-php .php
```

AddType application/x-httpd-php-source .phps

Örnek Kod:

```
<?php
// Connecting, selecting database
$dbconn = pg_connect("host=localhost dbname=deneme2 user=acikkod password=pass1"
    or die('Could not connect: ' . pg_last_error()));

// Performing SQL query
$query = 'SELECT * FROM test';
$result = pg_query($query) or die('Query failed: ' . pg_last_error());

// Printing results in HTML
echo "<table>\n";
while ($line = pg_fetch_array($result, null, PGSQL_ASSOC)) {
    echo "\t<tr>\n";
    foreach ($line as $col_value) {
        echo "\t\t<td>$col_value</td>\n";
    }
    echo "\t</tr>\n";
}
echo "</table>\n";

// Free resultset
pg_free_result($result);

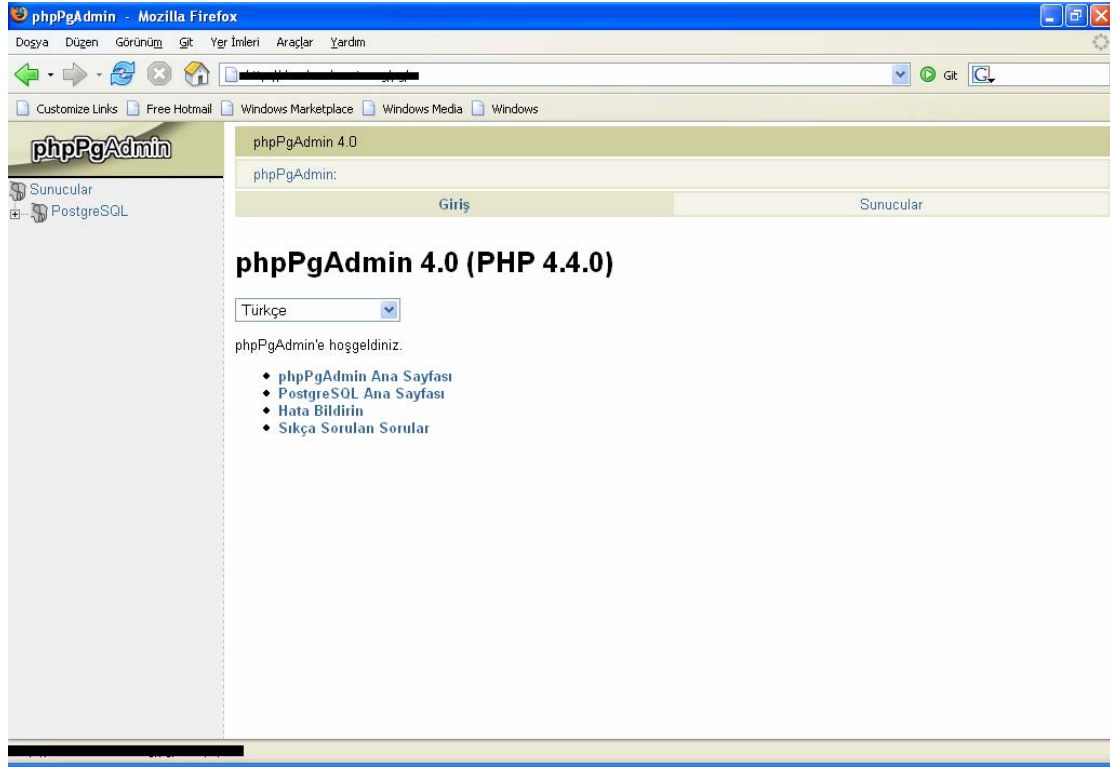
// Closing connection
pg_close($dbconn);
?>
```

6. phpPgAdmin Kurulumu

```
# cd /usr/ports/databases/phpPgAdmin/
# make PGADMDIR=www/data/phpPgAdmin install clean
```

Kurulumdan sonra phpPgAdmin'e <http://sunucuadi.com.tr/phpPgAdmin> Adresinden ulaşılabilir.

Not: Güvenlik sebebiyle pgsqk kullanıcısının giriş yapmasına izin verilmiyoruz.



7. Kaynaklar:

PostgreSQL El kitabı: <http://www.postgresql.org/docs/8.0/interactive/index.html>

<http://www.freebsdidiary.org/postgresql.php>

<http://www.PHP.net>

PostgreSQL man sayfaları