

# POSTGRESQL

## Veritabanı Sunucusu ve PHP Programlama Dili



Devrim GÜNDÜZ

[devrim@gunduz.org](mailto:devrim@gunduz.org)

Teknoloji Destek Merkezi

<http://seminer.linux.org.tr>

<http://www.linux.org.tr/belgeler.php>

<http://www.gunduz.org/seminer/pg>

*Bu sunu, herhangi bir izne gerek olmadan  
istenilen şekilde kullanılabilir.*

*(Kopyalama hakkı en-gel-le-ne mez! :-) )*

# LKD Seminerleri

- Seminer programı, seminer notları ve ayrıntılı bilgiler için;

<http://seminer.linux.org.tr>

adresini ziyaret edebilirsiniz.

# PostgreSQL nedir?

- PostgreSQL, veritabanları için relational modeli kullanan ve SQL standart sorgu dilini destekleyen bir veritabanı yönetim sistemidir.
- PostgreSQL aynı zamanda iyi performans veren, güvenli ve geniş özellikleri olan bir DBMS'tir. Hemen hemen tüm UNIX ya da Unix türevi (Linux, FreeBSD gibi) işletim sistemlerinde çalışır. Ayrıca NT çekirdekli tüm Windows sistemlerde de çalıştırılabilir. Tabii ki ücretsiz ve açık kodludur.

Kaynak: <http://www.postgresql.org>

# Kimler kullanıyor?

- Türkiye'deki belli başlı üniversiteler
- Malatya İnönü Üniversitesi Turgut Özal Tıp Merkezi
- [www.begendim.com](http://www.begendim.com)
- .info ve .org alan adları ([www.afilias.com](http://www.afilias.com))
- <http://www.sciencetunnel.com>
- Cisco
- BASF
- Red Hat
- Chrysler
- 3Com
- USDA, FDA, ve University of California
- ....

# Neden PostgreSQL?

- Açık kaynak kodlu
- Güvenilir
- Geniş geliştirici grubu
- 34 farklı platform desteği
- Bir çok programlama arayüzü
- Geniş destek imkanı
- Çok gelişmiş özellikler
- ...

# Teknik açıdan PostgreSQL

- Bazı teknik özellikler açısından, PostgreSQL şunları sunar:
  - %100 ACID uyumlu
  - ANSI SQL uyumlu
  - Referential Integrity
  - Replikasyon (ticari ve ticari olmayan çözümler) ana veritabanının (master) çok sayıda başka veritabanlarına (slave) çoklanmasını sağlar.

# Teknik açıdan PostgreSQL

- ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python, and Ruby için doğal arabirimler.
- Rules
- Views
- Triggers
- Sequences
- Inheritance
- Outer-Joins

# Teknik açıdan PostgreSQL

- Stored Procedures
- Kod geliştiriciler için açık API
- Doğal SSL Desteği
- UNION, UNION ALL ve EXCEPT sorgularına destek
- Doğal Kerberos Yetkilendirmesi
- Fonksiyonel ve Partial Indexler
- Procedural Diller



# Teknik açıdan PostgreSQL

- MD5, SHA1, XML ve diğer işlevsel özelliklerin yüklenebilmesi.
- Diğer SQL-uyumlu sistemlerle paylaşabilmek için taşınabilir SQL yaratan araçlar.
- Özelleştirilmiş, kullanıcı-tanımlı veri tipleri için geliştirilebilir veri tipi sistemi ve hızla gelişen yeni veri tipleri.
- Daha az SQL-uyumlu RDBMS'lerden geçişi kolaylaştıran cross-database uyumluluk fonksiyonları

# Desteklenen Platformlar

- Linux (Kernel 2.0.X ve üzeri)
- AIX 4.3.2
- HP – UX 9.0x ve 10.20
- FreeBSD 4.X
- IRIX 6.5.6f
- MacOS-X Darwin
- NetBSD 1.4, 1.4u
- QNX 4.25
- SCO OpenServer 5
- SCO Unix Ware 7
- Solaris 2.5.1-2.7
- Sun OS 4.1.14
- WinNT/Cygwin
- BSDI 4.0.1
- BeOS 5.0.3

# PostgreSQL Sınırlamaları

- Veritabanı için max büyüklük:
  - Bir tablo için max büyüklük:
  - Bir row için max büyüklük:
  - Bir field için max büyüklük
  - Tablo içindeki max row sayısı:
  - Tablo içindeki max column sayısı:
  - Tablo içindeki max index sayısı:
- Sınırsız
  - 64 TB (Tüm işletim sistemlerinde)
  - Sürüm 7.1 ve sonrasında sınırsız
  - Sürüm 7.1 ve sonrasında 1 GB
  - Sınırsız
  - 1600
  - Sınırsız

**Burada bahsedilen sınırlamalar, fiziksel sınırların haricindeki sınırlardır.**

# PostgreSQL

## Programlama Arayüzleri

- C
- PHP
- Java
- Python
- Delphi
- Perl
- Tcl
- Tk

# PostgreSQL araçları - psql

- Oracle'daki SQL\*PLUS gibi PostgreSQL'de psql adında command line aracı vardır. PostgreSQL veritabanları genellikle bu uygulama tarafından yaratılır ve yönetilir. psql :

**psql [seçenekler] [veritabanı\_adı [kullanıcı\_adı]]**

**biçiminde çalışır.**

# PostgreSQL araçları - psql

- `[postgres@localhost postgres] psql pgornek`
- Varsayılan veritabanı, kullanıcı adı, sunucu makine adı ve dinlenen port numarası sırasıyla PGDATABASE, PGUSER, PGHOST ve PGPORT çevre değişkenlerinin ayarlanması ile değiştirilebilir.
- Bu varsayılan değerler yine sırasıyla psql'e `-d`, `-U`, `-h` ve `-p` seçeneklerini geçirerek değiştirilebilir.

# PostgreSQL araçları - psql

- psql' e PostgreSQL'in desteklediği herhangi bir SQL komutunu verebilirsiniz. Desteklenen SQL komutlarının listesini \h iç komutu ile görebiliriz. Özel olarak istenen bir komut varsa \h sql\_komutu ile de yardımı alabiliriz. \? iç bize tüm iç komut listesini verecektir.

# PostgreSQL araçları - psql

- psql de komutlar birden fazla satırda yazılabilir. Böyle zamanlarda psql promptu > şekline dönüşecek ve daha fazla girişin beklendiği belirtilecektir:

```
[postgres@localhost postgres] /usr/local/pgsql/bin/psql pgornek
. . .
pgornek=> SELECT *
pgornek-> FROM muster_i
pgornek->;
...
[postgres@localhost postgres]
```



# Görsel Araçlar – phpPgAdmin

- Web arayüzlü olduğu için, çalıştırıldığı ortamdan bağımsızdır.
- postgres kullanıcısı ile veritabanınızı yönetebileceğiniz gibi belirli bir kullanıcı ile belirli bir veritabanını da yönetebilirsiniz.
- Her türlü kayıt ekleme, silme, değiştirme; veritabanı/tablo yaratma vb işlemlerini SQL bilmeden yapabilirsiniz.

# Görsel Araçlar – PHPPgAdmin

- Veritabanının içeriğini bir dosyaya boşaltabilir, daha sonra bu içeriği başka bir sunucuda da kullanabilirsiniz.
- Bunların dışında PostgreSQL'in SQL komutlarını çalıştırabilirsiniz.
- Türkçe dil desteği 2.4.2 sürümü ile gelmiştir.
- <http://phppgadmin.sourceforge.net> adresinden ücretsiz olarak indirilebilir.

# PHP ve PostgreSQL

- PHP, günümüzde çok fazla kullanılan, sunucu tarafında programlama amacıyla tasarlanmış bir dildir. Özellikle web üzerinde geliştirilen yazılımlarda kullanılır.
- PHP, özellikle 4.2 sürümü ile birlikte, oldukça fazla PostgreSQL fonksiyonu getirmiştir.
- <http://tr.php.net/manual/en/ref.pgsql.php>

# PHP ve PostgreSQL

- PHP, öntanımlı olarak PostgreSQL desteği ile gelmez. Bu desteği kendinizin vermesi gerekir.
  - Kaynak kod
  - RPM

# Derleme parametreleri

- Kaynak koddan derlemede, configure betiğine
  - with-pgsql=/var/lib/pgsql (eski rpm kurulumları)
  - with-pgsql=/usr/local/pgsql (eski kaynak koddan kurulumlar)
  - with-pgsql=/usr (PostgreSQL'in yeni sürümleri ve yeni PHP sürümleri) parametresi geçirilmelidir.

# Kaynak koddan derleme

- Kaynak koddan derlemede, sisteminize daha önceden PostgreSQL kurulu olmalıdır.

Bu şekilde, derleme için gerekli kitaplıklar bulunabilecektir.

# RPM Kurulumu

- php-pgsql paketi.
- rpm -ivh php-pgsql-<surum\_no>-<işlemci mimarisi>.rpm
- Örnek : rpm -ivh php-pgsql-4.2.2-1.i686.rpm

# PHP ve PostgreSQL

- RPM kurulumlarında, postgresql-devel paketinin sisteminize daha önceden kurulmuş olması gerekir.
- Bazı dağıtımlarda, php.ini dosyasını edit etmek gerekebilir.



# PHP PostgreSQL Fonksiyonları

- pg\_connect(), pg\_pconnect()
- pg\_query()
- pg\_close()
- pg\_update()
- pg\_last\_oid()
- pg\_field\_size()
- pg\_field\_name()
- pg\_connection\_busy()
- pg\_fetch\_array()
- pg\_field\_is\_null()
- pg\_free\_result()
- pg\_fetch\_result()
- pg\_lo\_open()
- pg\_set\_client\_encoding()
- pg\_ping()
- pg\_result\_error()
- pg\_select()
- pg\_affected\_rows()

# PHP PostgreSQL Fonksiyonları

- `pg_connect` : PostgreSQL bağlantısı açar:
  - `$conn = pg_connect ("dbname=deneme")`
    - *“deneme” veritabanına bağlantı.*
  - `$conn = pg_connect ("host= localhost dbname=deneme port=5432");`
    - *# “deneme” veritabanına “localhost” üzerinden “5432”.portu kullanarak bağlantı. TCP-IP portu açılmış olmalıdır.*
  - `$conn = pg_connect ("host= lkd dbname=deneme port=5437 user=postgres password=himm");`
    - *lkd adlı sunucudaki “deneme” adlı veritabanına “5432”.port üzerinden “postgres” kullanıcısı ve “himm” şifresi ile bağlantı.*

# PHP PostgreSQL Fonksiyonları

- `pg_close` : PostgreSQL bağlantısı kapatır:
  - Kullanımı :
  - **`bool pg_close (int connection)`**
- Örnek:

```
if (!@pg_close($conn))  
{  
    echo "Hata oluştu!!!";  
    exit;  
}
```

- `$conn` bağlantısını kapatır.

# PHP PostgreSQL Fonksiyonları

- `pg_query` : Sorguyu işletir:
  - Kullanımı :
  - **`int pg_query (int connection, string query)`**
- PostgreSQL veritabanına SQL sorgusu gönderir.
- Örnek:
- `$result = pg_query ($conn,$sorgu);`
  - Eğer sorguyu işletemezse elde edilecek hata mesajını `pg_ErrorMessage()` komutunu kullanarak görebilirsiniz.

# PHP PostgreSQL Fonksiyonları

- `pg_set_client_encoding` : İstemci tarafındaki encodingini ayarlar:
  - Kullanımı :
  - `int pg_set_client_encoding (int connection, string encoding)`
- Örnek:
- `$result = pg_query ($conn,'LATIN5');`
  - Türkçe için gerekli ayarlamaları yapar.

# PHP PostgreSQL Fonksiyonları

- `pg_client_encoding` : İstemci tarafındaki encoding değerini alır:
  - Kullanımı :
  - **`String pg_client_encoding (int connection);`**
- Örnek:
- `$result = pg_client_encoding ($conn);`

# PHP PostgreSQL Fonksiyonları

- `pg_connection_busy` : Bağlantının meşgul olup olmadığı bilgisini alır:
  - Kullanımı :
  - **`bool pg_connection_busy (resource connection);`**
- PostgreSQL veritabanına SQL sorgusu gönderir.
- Örnek:
- `$result = pg_connectionbusy ($conn);`

# PHP PostgreSQL Fonksiyonları

- `pg_fetch_all` : Sorgu sonucu gelen tüm sonuçları bir:
  - Kullanımı :
  - **`array pg_fetch_all (int result)`**
- `pg_fetch_all()` sonuç itibariyle gelen tüm satırları (row,recort,tuple) bir diziye atar. Eğer herhangi bir satır dönmemişse, hiç bir kayıt yok demektir.



# PHP PostgreSQL Fonksiyonları

- ÖRNEK:

```
<?php
```

```
    $conn = pg_pconnect("dbname=publisher");
```

```
    if (!$conn) {
```

```
        echo "An error occurred.\n";    exit;
```

```
    }
```

```
    $result = pg_query($conn, "SELECT * FROM authors");
```

```
    if (!$result) {
```

```
        echo "An error occurred.\n";
```

```
        exit;
```

```
    }
```

```
    $arr = pg_fetch_all($result);
```

```
    var_dump($arr);
```

```
?>
```

# PHP PostgreSQL Fonksiyonları

- `pg_last_oid` : Son oid'i alır:
  - **Kullanımı :**
  - **`int pg_last_oid (resource result).`**
- **Örnek:**
  - `function sql_last_inserted_id($connection, $result, $table_name, $column_name) {`
  - `$oid = pg_last_oid ( $result);`
  - `$query_for_id = "SELECT $column_name FROM $table_name WHERE`  
`oid=$oid";`
  - `$result_for_id = pg_query($connection,$query_for_id);`
  - `if(pg_num_rows($result_for_id))`
  - `$ids=pg_fetch_array($result_for_id,0,PGSQL_ASSOC);`
  - `return $ids[$column_name];`
  - `}`

# PHP PostgreSQL Fonksiyonları

- `pg_query` : Sorguyu işletir:
  - Kullanımı :
  - **`int pg_query (int connection, string query)`**
- PostgreSQL veritabanına SQL sorgusu gönderir.
- Örnek:
- `$result = pg_query ($conn,$sorgu);`
  - Eğer sorguyu işletemezse elde edilecek hata mesajını `pg_ErrorMessage()` komutunu kullanarak görebilirsiniz.

# PHP PostgreSQL Fonksiyonları

- `pg_query` : Sorguyu işletir:
  - Kullanımı :
  - **`int pg_query (int connection, string query)`**
- PostgreSQL veritabanına SQL sorgusu gönderir.
- Örnek:
- `$result = pg_query ($conn,$sorgu);`
  - Eğer sorguyu işletemezse elde edilecek hata mesajını `pg_ErrorMessage()` komutunu kullanarak görebilirsiniz.

# PHP PostgreSQL Fonksiyonları

- `pg_query` : Sorguyu işletir:
  - Kullanımı :
  - **`int pg_query (int connection, string query)`**
- PostgreSQL veritabanına SQL sorgusu gönderir.
- Örnek:
- `$result = pg_query ($conn,$sorgu);`
  - Eğer sorguyu işletemezse elde edilecek hata mesajını `pg_ErrorMessage()` komutunu kullanarak görebilirsiniz.

# PHP PostgreSQL Fonksiyonları

- `pg_query` : Sorguyu işletir:
  - Kullanımı :
  - **`int pg_query (int connection, string query)`**
- PostgreSQL veritabanına SQL sorgusu gönderir.
- Örnek:
- `$result = pg_query ($conn,$sorgu);`
  - Eğer sorguyu işletemezse elde edilecek hata mesajını `pg_ErrorMessage()` komutunu kullanarak görebilirsiniz.

# PHP PostgreSQL Fonksiyonları

- `pg_affected_rows` : Sorgu sonucu etkilenen satır sayısını verir :
  - Kullanımı :
  - **`int pg_affected_rows (int result_id)`**
- Örnek:
- ```
$sonuc = pg_query($conn,$sorgu);  
if(!pg_affected_rows($sonuc))  
{ # Sorgu sonucu etkilenen satır sayısı  
echo "HATA : Veritabanına kayıt girilemedi.<BR>";  
echo "<a href =\"_URL\">Geri</a><BR>";  
$sonuc = pg_query($conn, "abort"); # transaction iptali  
$sonuc = pg_query($conn, "rollback");  
pg_close($conn);  
exit;  
}
```

# PHP PostgreSQL Fonksiyonları

- `<?php`
- `$db = pg_connect ('dbname=foo');`
- `$data = array('field1'=>'AA', 'field2'=>'BB');`
- `// This is safe, since $_POST is converted automatically`
- `$res = pg_update($db, 'post_log', $_POST, $data);`
- `if ($res) {`
- `echo "Data is updated: $res\n";`
- `}`
- `else {`
- `echo "User must have sent wrong inputs\n";`
- `}`
- `?>`



# PHP ve PostgreSQL

## Formdan veri alma

- Önce formu hazırlayalım:
  - <FORM ACTION = insert.php METHOD=POST>
  - OPER ADI : <input type=text name=oper\_adi><BR>
  - OGRENCİ ADI : <input type=text name=kisi\_adi> <BR>
  - OGRENCİ NO : <input type=text name=id\_no><BR>
  - KİMLİK TÜRÜ : <input type=text name=kimlik\_turu><BR>
  - LAB : <input type=text name=lab><BR>
  - <input type=submit value = Gönder>
  - <input type=reset value= Temizle>
  - </FORM>

# PHP ve PostgreSQL

## --Veri Girme

- `<?php`
- `// Veritabanına bağlantı`
- `if (!$conn = @pg_connect("host=localhost dbname=deneme port=5432 user=postgres password=sifrem"))`
  - `{`
    - `die ("Veri tabanına bağlantı kurulamadı");`
  - `0`
  - `// Degerler`
- `$tarih = (date ("d-M-Y H:i:s"));`
- `$sorgu = "insert into kayip_kimlik (oper_adi,kisi_adi,id_no,kimlik_turu,tarih,lab) values`
- `(' $oper_adi', '$kisi_adi', '$id_no', '$kimlik_turu', '$tarih', '$lab')";`
-

# PHP ve PostgreSQL

## --Veri Girme

- `$result = pg_query("begin"); # transaction başlangıcı`
- `$sonuc = pg_query ($conn,$sorgu);`
- 
- `if(!$sonuc) {`
- `echo "HATA : Veri veritabanına girilemedi!!!<BR>";`
- `echo "<a href =\"form.html\">Geri</a><BR>";`
- `$sonuc = pg_query($conn, "abort"); # transaction iptali`
- `$sonuc = pg_query($conn, "rollback");`
- `pg_close($conn);`
- `exit;`
- `}`

# PHP ve PostgreSQL

## --Veri Girme

```
▪ if(!pg_affected_rows($sonuc)) { # Sorgu sonucu etkilenen satır sayısı
▪     echo "HATA : Veritabanına değer girilemedi!!!<BR>";
▪     echo "<a href =\"form.html\">Geri</a><BR>";
▪     $sonuc = pg_query($conn, "abort"); # transaction iptali
▪     $sonuc = pg_query($conn, "rollback");
▪     pg_close($conn);
▪     exit;
▪ }
▪ $result = pg_query("commit"); # transaction'ın işletiliyor
▪ $result = pg_query("end");    # transaction tamamlandı
▪
▪     echo "Ok. Kayıt Girildi.";
▪
▪ exit;
▪ ?>
```

# PHP ve PostgreSQL

- İf (!sonuc ....) Hata verirse SQL sorgusunun yazımında hata var demektir.
- İf (!pg\_affected\_rows(\$sonuc)) Hata verirse sorgudaki değerlerin en az birinde hata var demektir.

# PHP ve PostgreSQL

## --Veri Sorgulama

- <?PHP
- // Veritabanına bağlantı
- ...  
\$sorgu = "select \* from kayip\_kimlik order by oper\_adi";  
\$sonuc = pg\_query(\$conn,\$sorgu);  
if (pg\_num\_rows(\$sonuc)=='0') {die "Veritabanında kayıt yok...";}

# PHP ve PostgreSQL

## --Veri Sorgulama

```
• for ($i = 0; $i < pg_numrows($sonuc); $i++) {  
•     $satir = pg_fetch_row($sonuc, $i);  
•     $soper_adi = StripSlashes(trim($satir[0]));  
•     $kisi_adi = StripSlashes(trim($satir[1]));  
•     $sid_no = StripSlashes(trim($satir[2]));  
•     $kimlik_turu = StripSlashes(trim($satir[3]));  
•     $tarih = StripSlashes(trim($satir[4]));  
•     $lab = StripSlashes(trim($satir[5]));  
•     $k = $i+1; //veritabanindaki kayıt no  
•     echo "KAYIT NO : $k<BR>";  
•     echo "Oper Adı : $soper_adi<BR>";  
•     echo "Ogrenci adi :$kisi_adi <BR>";  
•     echo "Ogrenci no :$sid_no <BR>";  
•     echo "Kimlik Türü :$kimlik_turu <BR>";  
•     echo "Tarih :$tarih <BR>";  
•     echo "Lab :$lab <BR>";  
•     echo "<HR>";
```

# Kaynaklar

- <http://www.PostgreSQL.org>
- <http://advocacy.PostgreSQL.org>
- <http://techdocs.PostgreSQL.org>
- <http://www.commandprompt.com>
- <http://www.PostgreSQL.org/docs>
- <http://developer.PostgreSQL.org>



# Kaynaklar

- <http://www.pgsql.com/>
- <http://foundries.sourceforge.net/databases>
- <http://gborg.PostgreSQL.org>
- [www.wrox.com](http://www.wrox.com) - Beginning Databases with PostgreSQL, 2001

# E-Posta Listeleri

- [linux-programlama@linux.org.tr](mailto:linux-programlama@linux.org.tr)  
Veritabanları için tartışma listesi  
(üye olmak için, <http://liste.linux.org.tr> web arayüzünü kullanabilirsiniz.)  
[ppsql-tr-genel@PostgreSQL.org](mailto:ppgsql-tr-genel@PostgreSQL.org)  
(PostgreSQL Türkiye E-Posta Listesi . PostgreSQL üzerinde her türlü konu konuşulabilmektedir.)
- <http://www.PostgreSQL.org/lists.html>  
adresinden ulaşabileceğiniz tüm e-posta listeleri

# POSTGRESQL

## Veritabanı Sunucusu ve PHP Programlama Dili



Devrim GÜNDÜZ

[devrim@gunduz.org](mailto:devrim@gunduz.org)

Teknoloji Destek Merkezi

<http://seminer.linux.org.tr>

<http://www.linux.org.tr/belgeler.php>

<http://www.gunduz.org/seminer/pg>

*Bu sunu, herhangi bir izne gerek olmadan  
istenilen şekilde kullanılabilir.*

*(Kopyalama hakkı en-gel-le-ne mez! :-) )*