

PL/pgSQL SQL Yordamsal Dili

Devrim GÜNDÜZ

devrim@gunduz.org

devrim@PostgreSQL.org

Teknoloji Destek Merkezi

<http://www.tdmsoft.com>

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

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

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

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

12 Şubat 2004
Trabzon

Akademik Bilişim Konferansı
Karadeniz Teknik Üniversitesi



Linux Kullanıcıları Derneği Seminerleri

- Seminer programı, seminer notları ve ayrıntılı bilgiler için;

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

adresini ziyaret edebilirsiniz.

Linux Kullanıcıları Derneği

3. Linux ve Özgür Yazılım Şenliği

- Geçmiş şenlikler
- 13-16 Mayıs 2004 Ankara Milli Kütüphane,
- Eğitim seminerleri, yılın penguenleri, oyun turnuvası, hazine avı ...
- Konaklama ve ulaşım imkanları
- <http://senlik.linux.org.tr>
- senlik@linux.org.tr

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
- Tübitak UEKAE (Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü)
- Malatya İnönü Üniversitesi Turgut Özal Tıp Merkezi
- www.begendim.com
- .info ve .org alan adları (www.afilias.com)
- <http://www.sciencetunnel.com>
- Cisco
- BASF
- Red Hat
- Chrysler
- 3Com
- USDA, FDA, ve University of California
-

Yordamsal Dil Kavramı (Procedural Languages)

- Yordamsal dilleri, kurumsal veritabanlarda, SQL sorgulama dilin yetersiz kaldığı işlemler için kullanılmaktadır.
- SQL dili, veriyi sorgulayıp analiz etmek için en uygundur. Veri izerinde adım-adım işleme, veriye erişimi kısıtlama, veriye erişimi sistematik hale getirmek gibi işlemler için kullanılmaktadır.

Yordamsal Dillerin Kullanım Alanları

- Bir tabloyu satır satır tarayıp, bazı satırları değiştirmek gereksinimi varsa, bu işlem istemci tarafından yazılmış bir programla yapıldığında, her satır için sunucu-istemci veri alışverişi yapılmaktadır. Yordamsal dil ile yapıldığında ise sunucu tarafında daha hızlı bir biçimde yapılabilmetedir.
- Bir tabloya bağlı yordamsal dili ile yazılmış tetikleyici, tabloya veri eklenmeden ya da değiştirilmeden önce gereken tetkik ve değişiklikleri yapmak için kullanılmaktadır.

Yordamsal Dillerin Kullanım Alanları

- Bir tablonun bazı kolınlarına erişimini engellemek için kullanıcıya doğrudan veri sorgulama etkisi verilmeyip, yordamsal dille yazılmış bir fonksiyon ile yapılmaktadır.
- Veritabana, sorgularda kullanılmak üzere, yordamsal dili ile yazılmış yeni fonksiyonlar eklemektedir. Bu fonksiyonlar hem satır bazlı (substr, upper, lower gibi) hem de aggregate (max, min, sum, avg, count gibi) olabilir.

PL/SQL Nedir?

- Oracle – PL/SQL
 - Yapısal dillere ait özelliklerin standart SQL'e eklenmesi
 - Ada dili örnek alınarak tasarlanmıştır.

PL/pgSQL nedir?

- PL/SQL'e benzer
- Yüklenebilir bir dil
- SQL sorgularının ve programatik ifadelerin birlikte
 - Bir veritabanı sunucusunda kullanılması
 - Uzak bir yerden verinin sürekli olarak veritabanından istenmesi
 - Üstünde mantıksal işlemler yapmak isteyen istemci uygulamaları tarafından oluşturulan ağ ve iletişim yüklerinin gruplanması

PL/pgSQL'in Özellikleri

- ‘PL/pg”SQL”’ : PL/pgSQL kodunuz içinden direk olarak SQL dilini kullanabileceğinizi belirtir.
- PL/pgSQL içinden SQL kullanımı, programlarınızın gücünü, esnekliğini ve başarımını (performans) artırır.
- Bir PL/pgSQL kod bloğunda çoklu SQL ifadeleri çalıştırılırsa, bu ifadeler tek bir seferde çalışır; bu, tek bir ifadenin işlenmesinden farklıdır.
- Taşınabilirdir: Fonksiyonları, PostgreSQL Veritabanı Sunucusu’nun çalıştığı tüm platformlarda kullanılabilir.

PL/pgSQL ile neler yapılabilir?

- PL/pgSQL kodu ile :
 - PostgreSQL'in tüm veri tiplerine
 - Operatörlerine
 - Fonksiyonlarına ulaşabilirsiniz.

PL/pgSQL'i veritabanına yüklemek

- Programlama dilleri, veritabanları tarafından ancak bir veritabanı nesnesi olarak yaratıldığında kullanılabilir.
- Dolayısıyla, PL/pgSQL'i kullanmadan önce onu veritabanınıza yüklemelisiniz (ön tanımlı olarak sistemimize PL/pgSQL kuruludur).

PL/pgSQL'i veritabanına yüklemek

- Komut satırı: createlang
- İstemci uygulaması ile: CREATE FUNCTION
 - call_handler gereksinimi
- template1 veritabanına PL/pgSQL yüklenmesi

Createlang ile PL/pgSQL yüklenmesi

- Basit kullanım
 - `call_handler` dil yaratılması komut tarafından halledilir.
- `createlang plpgsql tdmtest;`
 - `-U`, `-h` gibi parametreler

CREATE LANGUAGE ile PL/pgSQL yüklenmesi

- CREATE FUNCTION ile call_handler yaratılmalıdır:
 - CREATE FUNCTION plpgsql_call_handler(
RETURNS OPAQUE AS
'/postgres_library_path/plpgsql.so' LANGUAGE 'C'
 - /usr/lib/pgsql/plpgsql.so - > RPM kurulumu
 - /usr/local/pgsql/lib/plpgsql.so -> Kaynak kod kurulumu
-

CREATE LANGUAGE ile PL/pgSQL yüklenmesi

- tdmtest=# **CREATE FUNCTION plpgsql_call_handler ()**
tdmtest-# **RETURNS OPAQUE**
tdmtest-# **AS '/usr/local/pgsql/lib/plpgsql.so'**
tdmtest-# **LANGUAGE 'C';**
CREATE
- tdmtest=# **CREATE LANGUAGE 'plpgsql' HANDLER**
plpgsql_call_handler LANCOMPILER 'PL/pgSQL';
WARNING: changing return type of function plpgsql_call_handler
from "opaque" to "language_handler"
CREATE LANGUAGE

CREATE LANGUAGE ile PL/pgSQL yüklenmesi

- HANDLER anahtar sözcüğünü takip eden ad, call handler'ın yaratıldığı zaman kullanılan ad ile aynı olmalıdır. plpgsql_call_handler adıyla bir call handler yaratıldığı için, son işlemde de aynı ad kullanılmıştır.
- LANCOMPILER anahtar kelimesini takip eden kısım PostgreSQL 7.1.x ve sonrası için gerekli bir bölümdür. Genelde, dili tanımlamak için yorum alanı olarak kullanılır.

PL/pgSQL Dil Yapısı

- Basit yapı
 - Kodun her bir bölümü bir fonksiyon olarak çalışması için tasarlanmıştır.
- C diline benzer yapı
 - Kodun fonksiyonlarla yazılması
 - Değişkenlerin kullanılmadan önce declare edilmesi
- Büyük-küçük harfe duyarlı **değil**.
- " işareti

PL/pgSQL'de kod blokları

- Block structured code
- CREATE FUNCTION içine girilir.
 - Fonksiyon adı ve argüman tipleri
 - Ana blok
 - Declaration
 - Begin
 - ...

PL/pgSQL'de kod blokları

DECLARE

- Declaration
 - DECLARE anahtar sözcüğü
 - Seçimsel ilk değerler
 - Değişkenin adı ve tipi
 - Noktalı virgül ile bitiş

PL/pgSQL'de kod blokları

BEGIN - END

- Ana gövde
 - BEGIN anahtar sözcüğü
 - RETURN
 - Bütün alt bloklar END'den önce sonlanmalıdır.

PL/pgSQL kod bloğunun örnek yapısı

-

```
CREATE FUNCTION identifier (arguments) RETURNS type AS '  
  DECLARE  
    declaration;  
    [...]  
  BEGIN  
    statement;  
    [...]  
  END;  
' LANGUAGE 'plpgsql';
```

Alt bloklar (sub-blocks)

- PL/pgSQL kod bloğu sınırsız sayıda alt blok (sub-block) içerebilir. Alt bloklar, normal bloklarla aynı şekilde okunur ve yorumlanırlar; dolayısıyla onlar da kendi içlerinde alt blok içerebilirler...

Alt bloklar (sub-blocks)

- Alt bloklar büyük PL/pgSQL fonksiyonlarının düzenlenmesinde yardımcı olmaktadır. Tüm alt bloklar normal blok yapısını kullanırlar; yani DECLARE anahtar sözcüğü ile başlayıp BEGIN ile devam ederler; ifadelerin olduğu kısımdan sonra da END anahtar kelimesi ile biterler.

PL/pgSQL'de Yorumlar (Comments)

- İki yol
 - - Tek satırlı yorumlar
 - - Blok yorumlar (çok satırlı yorumlar)
- Programlama dillerinin yorum yapısına benzerler.

PL/pgSQL'de Yorumlar (Comments)

Tek satırlı yorumlar

- Yorum satırı yazmanın ilk yöntemi tek satırlı olanlardır.
- Tek satırlı yorumlar iki çizgi ile (--) başlarlar.
- Satır sonu karakteri barındırmazlar.

Örnek:

-- Bu satır, tek satırlı yorum olarak işlenir.

PL/pgSQL'de Yorumlar (Comments)

Blok yorumlar

- Blok yorumlar /* ile başlarlar.
- Birden fazla satır içerirler.
- */ ile biterler.

```
/*  
 * Bu bir  
 * blok yorum  
 * örneğidir.  
*/
```

PL/pgSQL'de Yorumlar

- Tek satırlı yorumlar blok yorumlar içinde kullanılabilirler; ancak blok yorumlar başka blok yorumlar içinde kullanılamazlar.

İfadeler ve Expression (Statements and Expressions)

- PL/pgSQL kodu ifadeler ve tanımlardan oluşur.
 - Ağırlık ifadelerdir.
 - Diğer programlama dillerine benzerler.

PL/pgSQL'de ifadeler

- İfadeler PL/pgSQL kodu içinde işlemler yaparlar:
 - Değişkene bir değer atanması
 - Bir sorgunun çalıştırılması
 - ...
- Her ifade ; işareti ile biter.

PL/pgSQL'de Expression

- Expressionlar, PostgreSQL'in temel veri tiplerinden birini sonucu olarak geri döndüren işlemler ve hesaplamalardır.
- Örnek: $a := b + c;$
b ve c değişkenlerini toplar ve bunu a değişkenine atar.

PL/pgSQL'de Expression - Örnek

- ```
CREATE FUNCTION bir_fonksiyon () RETURNS int4
AS '
DECLARE
 bir_tamsayi int4;
BEGIN
 bir_tamsayi := 10 * 10;
 RETURN bir_tamsayi;
END;
' LANGUAGE 'plpgsql';
```

# PL/pgSQL'de Expression - Örnek

- tdmtest=# SELECT bir\_fonksiyon() AS  
sonuc;  
sonuc  
-----  
100  
(1 row)

# Dinamik Sorgular

- Dinamik sorgular (EXECUTE anahtar kelimesi ile çalıştırılan SQL sorguları) dışında, bir fonksiyondaki tüm PL/pgSQL expressionları PostgreSQL backendinin yaşama süresi içinde sadece bir kez prepare edilir.
-

# Dinamik Sorgular

- Bu nedenle sabit değerler (*now* ve *current timestamp* sabitleri, sabit değişkenler değil) PL/pgSQL expressionları bir kez prepare edilir. Bu da run-time anında alınması gereken değerler içeren kodların sorun yaşamasına neden olur. Sorunun çözümü, PL/pgSQL 'in sabit timestamp değerleri fonksiyonun çalışma anında almasını sağlamaktır:

# Dinamik Sorgular

## Örnek

- `CREATE FUNCTION add_shipment (integer, text)`  
`RETURNS timestamp AS '`  
`DECLARE`

`-- Fonksiyon argumanları için alias declare bolumu`

`musteri_no ALIAS FOR $1;`

`isbn ALIAS FOR $2;`

# Dinamik Sorgular

## Örnek

- - Declare a variable to hold the shipment ID number
  - and the current time.
- shipment\_id INTEGER;
- right\_now timestamp;

# Dinamik Sorgular

## Örnek

- BEGIN
  - Set the current time variable to the string "now".  
right\_now := "now";
  - Order the existing shipments by their ID numbers,  
beginning
  - with the highest number, then insert the first ID  
number into
  - the shipment\_id variable.  
SELECT INTO shipment\_id id FROM shipments  
ORDER BY id DESC;

# Dinamik Sorgular

## Örnek

- -- Add one to the shipment\_id variable.  
shipment\_id := shipment\_id + 1  
/\* Insert a shipment record into the shipments table.  
The right\_now variable will be typecast to a timestamp  
at run-time, causing constant value now to be interpreted  
as the timestamp each time the function is run. \*/
- INSERT INTO shipments VALUES ( shipment\_id,  
customer\_id, isbn, right\_now );



# Dinamik Sorgular

## Örnek

- -- Return a timestamp using the constant value now.  
RETURN right\_now;  
END;  
' LANGUAGE 'plpgsql';

# PL/pgSQL'de değişken kullanımı

- PL/pgSQL'de değişken kullanımı
  - Düzenlenebilecek verilerin açıkça belirtilmiş veri tipinde saklanması
  - DECLARE anahtar kelimesinden sonra belirtilirler.
  - Ön tanımlı değer belirtilebilir.
    - Aksi taktirde NULL
    - FOR döngüsündeki değerın istinası

# PL/pgSQL'de değişken kullanımı :

## Veri Tipleri

- Herhangi bir SQL standart veri tipinde gösterilebilirler:
  - CHAR, INTEGER, BOOLEAN, DATE, vb.
- Ek olarak, RECORD veri tipi
  - Verinin değişkene 'girildiği' zamanki kolonları belirtmeden satır bilgisini tutmak için tasarlanmıştır.

# PL/pgSQL'de değişken kullanımı : Tanımlama (declaration)

- Değişkenlerin PL/pgSQL kod bloğunun içinde kullanılabilmesi için, kod bloğunun başında DECLARE anahtar kelimesinden sonra belirtilmesi gerekmektedir.
  - Tüm alt bloklarda geçerlidir.
  - Alt bloklardaki değişkenler o alt blokun tamamlanması ile sıfırlanırlar.

# PL/pgSQL'de değişken kullanımı : Örnek

- CREATE FUNCTION *identifier* (*arguments*)  
RETURNS *type* AS '  
DECLARE  
    -- Declare an integer.\*  
subject\_id INTEGER;  
    -- Declare a variable length character.  
book\_title VARCHAR(10);  
    -- Declare a floating point number.  
book\_price FLOAT;

# PL/pgSQL'de değişken kullanımı : Örnek (devam)

- BEGIN  
    *ifadeler*  
END;  
' LANGUAGE 'plpgsql'

# PL/pgSQL'de değişken kullanımı :

## Ek bilgiler

- CONSTANT anahtar kelimesi
  - Sabit değer belirtir.
- NOT NULL anahtar kelimesi
  - Değişkenin NULL olamayacağını belirtir.
- DEFAULT anahtar kelimesi
  - Değişkene ön tanımlı bir değer atar.
- *değişken adı* [ CONSTANT ] *veri\_tipi* [ NOT NULL ] [ { DEFAULT | := } *değer* ] ;

# PL/pgSQL'de değişken kullanımı :

## Değişken tanımlama

- `CREATE FUNCTION example_function () RETURNS text AS '  
DECLARE  
-- Declare a constant integer with a  
-- default value of 5.  
five CONSTANT INTEGER := 5;  
-- Declare an integer with a default  
-- value of 100 that cannot be NULL.  
ten INTEGER NOT NULL := 10;`



# PL/pgSQL'de değişken kullanımı :

## Değişken tanımlama

- -- Declare a character with  
-- a default value of "a".  
letter CHAR DEFAULT "a";

```
BEGIN
return letter;
END;
' LANGUAGE 'plpgsql';
```

# PL/pgSQL'de değişken kullanımı : Değer Atama

- DECLARE bölümünde yapılır.
  - := işareti kullanılır (assignment operator)

# PL/pgSQL'de değişken kullanımı : Değer Atama Örneği

- CREATE FUNCTION *identifier* (*arguments*)  
RETURNS *type* AS '  
    DECLARE  
        an\_integer int4 := 10;  
    BEGIN  
        *statement*;  
        [...]  
    END;  
' LANGUAGE 'plpgsql';

# 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@liste.linux.org.tr**

**Veritabanları için tartışma listesi  
(üye olmak için, <http://liste.linux.org.tr> web arayüzünü kullanabilirsiniz.)**

- **pgsql-tr-genel@PostgreSQL.org**

**(PostgreSQL Türkiye E-Posta Listesi . PostgreSQL üzerinde her türlü konu konuşulabilmektedir.)**

# PL/pgSQL SQL Yordamsal Dili

Devrim GÜNDÜZ

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

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

Teknoloji Destek Merkezi

<http://www.tdmsoft.com>



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

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

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

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