

İsim

CREATE FUNCTION – yeni bir işlev tanımlar

KULLANIM

```
CREATE [OR REPLACE] FUNCTION isim ( [ [ arg_ismi ] arg_türü [, ...] ] )
    RETURNS dönüş_türü
{ LANGUAGE dil_ismi
  | IMMUTABLE | STABLE | VOLATILE
  | CALLED ON NULL INPUT | RETURNS NULL ON NULL INPUT | STRICT
  | [ EXTERNAL ] SECURITY INVOKER | [ EXTERNAL ] SECURITY DEFINER
  | AS 'tanım'
  | AS 'nesne_dosyası' , 'ilintileme_sembolü'
} ...
[ WITH ( öznitelik [, ...] ) ]
```

Açıklama

CREATE FUNCTION yeni bir işlev tanımlar. **CREATE OR REPLACE FUNCTION** ise ya yeni bir işlev oluşturur ya da bir mevcut işlev tanımını değiştirir.

Eğer bir şema ismi belirtilmişse, işlev belirtilen şemada oluşturulur. Aksi takdirde, geçerli olan şemada oluşturulur. Yeni işlevin ismi, aynı şema içindeki aynı argüman türlerine sahip bir işlev ismi ile aynı olmalıdır. Farklı argüman türlerine sahip aynı isimdeki işlevlere izin verilir.

Mevcut bir işlevin tanımını güncellemek için **CREATE OR REPLACE FUNCTION** kullanılır. Ancak, bir işlevin ismini ya da argüman türlerini bu yöntemle değiştirmek mümkün değildir (eğer denerseniz, aslında ayrı ve yeni bir işlev oluşturmuş olursunuz). Ayrıca, **CREATE OR REPLACE FUNCTION** ile mevcut bir işlevin dönüş türünü değiştirmeye çalışmayın, bunu yapmak için işlevi önce silmeli, sonra yeniden oluşturmalısınız.

Eğer bir işlevi silip yeniden oluştursanız, yeni işlev eskisiyle aynı olmaz; mevcut kuralları, sanal tabloları, tetikler, vs. de silmiş olursunuz. Bunları koruyarak işlev tanımını değiştirmek için **CREATE OR REPLACE FUNCTION** kullanın.

İşlevi tanımlayan kullanıcı aynı zamanda sahibi olur.

Parametreler

isim

Tanımlanacak işlevin ismi (şema nitelemeli olabilir).

arg_ismi

Bir argümanın ismi. Bazı diller (şimdilik sadece PL/pgSQL) argüman ismini işlev gövdesinde kullanmanızı ister. Diğer diller için argüman ismi sadece ek bir bilgidir.

arg_türü

İşlevin (varsa) argümanlarının veri türleri (şema nitelemeli olabilir). Argüman türleri temel, birleşik veya veri alanı türü olabileceği gibi bir tablo sütununun veri türüne başvurulabilir.

Gerçekleşim diline bağımlılık sebebiyle, *cstring* gibi tanımlı türler de belirtilebilir. Argüman türü olarak tanımlı türler kısmen olabileceği gibi SQL veri türlerinin tamamen dışında türler olabilirler.

Bir sütun türüne başvuruyu *tablo_ismi.sütun_ismi*%TYPE yazarak belirtilebilirsiniz. Bu özelliğin kullanımı bazan bir işlevin bir tablo tanımındaki değişikliklerden bağımsız olmasını sağlayabilir.

dönüş_türü

İşlevin dönüş değerinin veri türü (şema nitelemeli olabilir). Dönüş türü temel, birleşik veya veri alanı türü olabileceği gibi bir tablo sütununun veri türüne başvuru olabilir.

SETOF değiştiricisi işlevin tek bir öge ile değil, bir öge kümesi ile döneceğini belirtir.

Bir sütun türüne başvuruyu *tablo_ismi.sütun_ismi*%TYPE yazarak belirtebilirsiniz.

dil_ismi

İşlevin gerçekleştirildiği dilin ismi. SQL, C, internal veya kullanıcı tarafından belirtilecek yordamsal dillerden biri olabilir. Geriye uyumluluk için isim tek tırnak içine alınabilir.

IMMUTABLE

STABLE

VOLATILE

Çalışma anı eniyilemesi için işlevin çoklu değerlendirmelerinin tek bir değerlendirme ile değiştirilmesinin güvenli olup olmayacağını belirlerler. En fazla bir seçim belirtilebilir. Bunlardan hiçbirini belirtilmemişse **VOLATILE** öntanımlıdır.

IMMUTABLE işlevin aynı argüman değerleriyle daima aynı sonucu döndüreceğini belirtir; yani, argümanlarının veritabanındaki değerlerden faydalanmayacağını ya da argüman olarak açıkça belirtilmemiş bilgileri kullanmayacağını belirtir. Eğer bu seçenek belirtilmişse, işlevin tamamen sabit argümanlı her çağrısı anında işlev değeriyle değiştirilebilir.

STABLE tek bir tablo taraması içinde işlevin aynı argüman değerleri için tutarlı olarak aynı sonucu döndüreceğini, ama sonucun SQL cümleleriyle değişebileceğini belirtir. Bu, sonuçları veritabanı aramalarına, parametrelerine, v.s. bağımlı işlevler için un uygun seçimdir. Ayrıca, değerleri bir hareket içinde değişmediğinden *current_timestamp* ailesindeki işlevler de **STABLE** olarak nitelenmiştir.

VOLATILE işlevin değerinin tek bir tablo taraması içinde bile değişebileceğini, dolayısıyla hiçbir eniyileme yapılamayacağını belirtir. Görece birkaç veritabanı işlevi bu anlamda oynaktır; bazı örnekler: *random()*, *currval()*, *timeofday()*. Yan etkiye sahip her işlevin, işlev sonucu tahmin edilebilir olsa bile, eniyilenmesini önlemek için **VOLATILE** olarak sınıflanması gerektiğini unutmayın; *setval()* buna bir örnektir.

Ek ayrıntılar için

<http://www.postgresql.org/docs/8.0/static/xfunc-volatility.html> adresine bakınız.

CALLED ON NULL INPUT

RETURNS NULL ON NULL INPUT

STRICT

CALLED ON NULL INPUT (öntanımlıdır) işlevin bazı argümanları NULL olduğunda normal olarak çağrılacağını belirtir. Bundan sonra işlev yazarının yapacağı NULL değerleri sınamak ve gereğini yapmaktır.

RETURNS NULL ON NULL INPUT veya **STRICT** ise işlevin, argümanlarından herhangi birinin NULL olması durumunda daima NULL değer döndüreceğini belirtir. Eğer bu seçenek belirtilmişse, NULL argümanların varlığında işlev çalıştırılmayacak, onun yerine hemen bir NULL değer döndürülecektir.

[**EXTERNAL**] **SECURITY INVOKER**

[**EXTERNAL**] **SECURITY DEFINER**

SECURITY INVOKER işlevin kendisini çağıran kullanıcının izinleriyle çalıştırılacağını belirtir. **SECURITY DEFINER** ise işlevin kendisini oluşturan kullanıcının izinleriyle çalıştırılacağını belirtir.

EXTERNAL sözcüğü SQL uyumluluğu için varsa da isteğe bağlı olmasından dolayı, SQL'in tersine bu özellik harici işlevlere uygulanmaz.

tanım

İşlevi tanımlayan bir dizge sabiti; anlamlandırılması dile bağımlıdır. Bir dahili işlev ismi olabileceği gibi bir nesne dosyasının yolu, bir SQL komutu veya bir yordamsal dilde yazılmış bir metin olabilir.

nesne_dosyası, ilintileme Sembolü

AS sözcüğünün bu biçimi özdevimli yüklenebilen C dili işlevleri için, işlevin C dili kaynak kodundaki ismi, bir SQL işlevinin ismiyle aynı değilse, kullanılır. *nesne_dosyası* dizgesi ile özdevimli yüklenebilen nesneyi içeren dosyanın ismi belirtilir. *ilintileme Sembolü* ise işlevin C dili kaynak kodundaki ismi olarak işlevin ilintileme sembolüdür. Eğer ilintileme sembolü belirtilmezse, tanımlandığı SQL işlevinin ismiyle aynı olduğu kabul edilir.

öznitelik

İşlev hakkında bilgi içeren isteğe bağlı parçaları belirtmenin tarihi bir yolu. Burada belirtilebilecek öznitelikler şunlardır:

`isStrict`

STRICT veya **RETURNS NULL ON NULL INPUT**'a eşdeğerdir.

`isCachable`

IMMUTABLE'ın atıl olmuş eşdeğeridir; hala geriye uyumluluk adına kabul edilmektedir.

Öznitelik isimleri harf büyüklüğüne duyarlıdır.

Ek Bilgi

İşlevlerin yazılması hakkında bilgi edinmek için

<http://www.postgresql.org/docs/8.0/static/xfunc.html> adresine bakınız.

Girdi argümanları ve dönüş değeri için tamamen **SQL** türü sözdizimine izin verilmektedir. Ancak, temel işlev gerçekleştiriminin sorumlusu olduğu tür belirtiminin bazı ayrıntıları (sayısal türlerdeki hassasiyet alanı gibi) **CREATE FUNCTION** tarafından sessizce sineye çekilir (yani, ne tanınır ne de uygulanır).

PostgreSQL, argümanları farklı olmak şartıyla işlevler için aynı isimlerin kullanımına izin verir. Ancak, C dilinde tüm işlev isimlerinin farklı olması gerekir, bu nedenle bu tür C işlevlerine farklı C isimleri vermelisiniz. (örneğin, C isimlerinin parçası olarak argüman türleri kullanılabilir).

CREATE FUNCTION aynı nesne dosyası için defalarca çağrıldığında dosya sadece bir kere yüklenir. Dosyayı tekrar tekrar yüklemeniz gerekiyorsa (geliştirme amaçlı) **LOAD [load(7)]** komutunu kullanın.

Kullanıcı tanımlı işlevleri kaldırmak için **DROP FUNCTION [drop_function(7)]** kullanılır.

İşlev tanımlarını yazarken tek tırnak içine almak yerine dolar işaretleri arasında yazmak çoğunlukla daha çok tercih edilen bir yöntemdir

(<http://www.postgresql.org/docs/8.0/static/sql-syntax.html#SQL-SYNTAX-DOLLAR-QUOTING> adresine bakınız).

Dolar işaretleri arasında yazma yöntemi kullanılmadığı zaman, işlev tanımı içindeki tek tırnaklar veya tersbölülerin öncelenmesi gerekir.

Bir işlevin tanımlanabilmesi için kullanıcının dil üzerinde **USAGE** iznine sahip olması gerekir.

Örnekler

Burada, başlangıç olarak yardımcı olacak önemsiz bir örneğe yer verilmiştir. Daha fazla bilgi edinmek ve başka örnekler için

<http://www.postgresql.org/docs/8.0/static/xfunc.html> adresine bakınız.

```
CREATE FUNCTION add(integer, integer) RETURNS integer
AS 'select $1 + $2;'
LANGUAGE SQL
IMMUTABLE
RETURNS NULL ON NULL INPUT;
```

PL/pgSQL dilinde bir argüman ismi kullanarak bir tamsayı arttırımı:

```
CREATE OR REPLACE FUNCTION increment(i integer) RETURNS integer AS $$
BEGIN
    RETURN i + 1;
END;
$$ LANGUAGE plpgsql;
```

Uyumluluk

SQL:1999 ve sonrasında bir **CREATE FUNCTION** tanımlıdır. PostgreSQL sürümü de benzer olmakla birlikte tamamen uyumlu değildir. Ne özellikler taşınabilir ne de farklı diller kullanılabilir.

İlgili Belgeler

ALTER FUNCTION [[alter_function\(7\)](#)], **DROP FUNCTION** [[drop_function\(7\)](#)], **GRANT** [[grant\(7\)](#)], **LOAD** [[load\(7\)](#)], **REVOKE** [[revoke\(7\)](#)], **createlang(1)**.

Çeviren

Nilgün Belma Bugüner <[nilgun \(at\) belgeler-gen-tr](mailto:nilgun (at) belgeler-gen-tr)>, Nisan 2005

YASAL UYARI

Bu çevirinin telif hakkı yukarıda belirtilen çevirmen(ler)e aittir. Özgün belgenin telif hakkı ve lisans bilgileri varsa ve belge içinde belirtilmemişse belge sonunda belirtilmiş olacaktır. Bu çevirinin lisansı, özgün belge için belirtilmiş bir lisans varsa ve bu lisans çevirinin de aynı lisansa sahip olmasını gerektiriyorsa onunla aynıdır, yoksa GNU GPL lisansı ve her iki durumda da ek olarak aşağıdaki koşullar geçerlidir. GNU GPL lisansı <<http://www.gnu.org/licenses/gpl.html>> adresinden edinilebilir.

BU BELGE ÜCRETSİZ OLARAK RUHSATLANDIĞI İÇİN, BELGENİN İÇERDİĞİ BİLGİLERİN VEYA KODLARIN NİTELİKLERİ İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGELERİ "OLDUĞU GİBİ", AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BELGELERİN KALİTESİ VEYA PERFORMANSI İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATA VEYA EKSİKLİKTEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE BELGEYİ DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BELGENİN İÇERDİĞİ BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİNİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHİ, SORUMLU DEĞİLDİR.

PostgreSQL

CREATE FUNCTION(7)

Bu dosya (man7-create_function.pdf), belgenin XML biçiminin \TeX Live ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

31 Ocak 2007