

İsim

CREATE TRIGGER – yeni bir tetik tanımlar

KULLANIM

```
CREATE TRIGGER isim { BEFORE | AFTER } { olay [ OR ... ] }  
ON tablo [ FOR [ EACH ] { ROW | STATEMENT } ]  
EXECUTE PROCEDURE işlev_ismi ( argümanlar )
```

Açıklama

CREATE TRIGGER yeni bir tetik oluşturur. Tetik belirtilen tablo ile ilişkilendirilecek ve belli bir olay meydana geldiğinde *işlev_ismi* ile belirtilen işlevi çalıştıracaktır.

Tetiğin ya bir satır üzerinde işlem yapılmaya çalışılmadan önce (kısıtlar sınanmadan ve **INSERT**, **UPDATE** veya **DELETE** yapılmadan önce) ya da işlem tamamlandıktan sonra (kısıtlar sılandıktan ve **INSERT**, **UPDATE** veya **DELETE** tamamlandıktan sonra) çalışacağı belirtilebilir. Eğer tetik olaydan önce çalışırsa, geçerli satır için işlemi atlayabilir ya da veri girilen satır değişebilir (sadece **INSERT** ve **UPDATE** işlemleri için). Eğer tetik olaydan sonra çalışırsa, tüm değişiklikler, son veri girme, güncelleme veya silme işlemi tetiğe görünür olur.

FOR EACH ROW imli bir tetik işlemi değiştiren her satır için bir defa çağrılır. Örneğin, 10 satırı etkileyen bir **DELETE**, her satır silinişinde bir kere olmak üzere 10 ayrı defa **ON DELETE** tetiğinin çağrılmasına sebep olur. Tersine, **FOR EACH STATEMENT** imli bir tetik belirtilen bir işlem için işlemin kaç satırı etkilediğinden bağımsız olarak, sadece bir defa çalıştırılır (hatta, işlem hiçbir satırı değiştirmese bile tetik yine de çalıştırılacaktır).

Eğer aynı olay için aynı türden çok sayıda tetik tanımlanmışsa, bunlar isimlerine göre alfabetik sırayla çalıştırılırlar.

SELECT herhangi bir satırı değiştirmedikten, **SELECT** tetikleri oluşturamazsınız. Kurallar ve sanal tablolar böyle durumlarda daha uygundur.

Tetikler hakkında daha fazla bilgi edinmek için

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

Parametreler

isim

Yeni tetiği belirtecek isim. Bu isim, aynı tablodaki diğer tetiklerin isimlerinden farklı olmalıdır.

BEFORE

AFTER

İşlevin olaydan önce mi sonra mı çağrılacağını belirler.

olay

INSERT, **UPDATE** veya **DELETE**'den biri; tetiği çalıştıracak olayı belirtmek için kullanılır. **OR** kullanarak çok sayıda olay belirtilebilir.

tablo

Tetiği içerecek tablonun ismi (şema nitelemeli olabilir).

FOR EACH ROW

FOR EACH STATEMENT

Tetik işlevinin tetikleyici olay tarafından her satır için bir kere mi yoksa her SQL deyimi için bir kere mi çalıştırılacağını belirler. Eğer hiçbiri belirtilmemişse, **FOR EACH STATEMENT** öntanımlıdır.

işlev_ismi

Tetikle çalıştırılan, kullanıcı tarafından argümansız ve dönüş türü **trigger** olarak bildirilmiş ve tanımlanmış bir işlev.

argümanlar

Tetik tarafından çalıştırıldığında işleve aktarılacak isteğe bağlı argümanların virgülle ayrılmış listesi. Argümanlar dizge sabitlerdir. Basit isimler ve sayısal sabitler de burada yazılabilir, fakat her biri dizgeye dönüştürülecektir. Tetikleyici işlev dilinin nasıl gerçekleştirildiğini incelemeniz ve işlev içinde tetik argümanlarına nasıl erişildiğini öğrenmeniz önerilir; normal işlev argümanlarından farklı ele alınıyor olabilirler.

Ek Bilgi

Bir tablo içinde bir tetik oluşturmak için, kullanıcı tablo üzerinde **TRIGGER** yetkisine sahip olmalıdır.

7.3 öncesi PostgreSQL sürümlerinde, işlevin dönüş türü olarak **trigger** değil, **opaque** bildiriliyordu. Eski sürüme ait döküm dosyalarını desteklemek için, **CREATE TRIGGER opaque** türünde değer döndüren işlevleri kabul edecek, ama işlevin bildirilen dönüş türünün **trigger** olarak değiştirilmesini belirten bir uyarı alacaksınız.

Bir tetiği silmek için **DROP TRIGGER** [`drop_trigger(7)`] kullanın.

Örnekler

<http://www.postgresql.org/docs/8.0/static/triggers.html> adresinde eksiksiz bir örnek bulacaksınız.

Uyumluluk

CREATE TRIGGER cümlesinin PostgreSQL gerçeklenimi SQL:1999 standardının bir alt kümesini içerir. (SQL-92'de tetikler için bir madde yoktur.) Mevcut olmayan işlevsellik şöyle sıralanabilir:

- SQL:1999 belli sütunların güncellenmesiyle çalıştırılacak tetiklere izin verir (**AFTER UPDATE OF col1, col2** gibi).
- SQL:1999 "eski" ve "yeni" satırlar için ya da tetikleyen eylemin (**CREATE TRIGGER ... ON tabloismi REFERENCING OLD ROW AS birisim NEW ROW AS yeniisim ...** gibi) tanımında kullanmak için tablolara takma isim tanımlamanıza izin verir. PostgreSQL tetik işlevlerinin kullanıcının seçimine bağlı olarak herhangi bir dilde yazılmasına izin verdiğinden veriye erişim seçilen dile özel yollarla olur.
- PostgreSQL sadece tetikleyen eylem için kullanıcı tanımlı bir işlevin çalıştırılmasına izin verir. SQL:1999 ise, tetiklenen eylem olarak **CREATE TABLE** gibi başka SQL cümlelerinin çalıştırılmasına izin verir. Bu sınırlamanın, istenen SQL cümlelerini çalıştıracak kullanıcı tanımlı bir işlev oluşturarak çevresinden dolanmak hiç de zor değildir.

SQL:1999, çok sayıda tetiğin oluşturulma zamanı sırasına göre çalıştırılmasına izin verir. PostgreSQL ise çalışılması daha uygun olan isim sıralamasını kullanır.

İlgili Belgeler

CREATE FUNCTION [`create_function(7)`], **ALTER TRIGGER** [`alter_trigger(7)`], **DROP TRIGGER** [`drop_trigger(7)`].

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 DAHI, SORUMLU DEĞİLDİR.

PostgreSQL

CREATE TRIGGER(7)

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

31 Ocak 2007