

## İsim

CREATE AGGREGATE – yeni bir ortak değer işlevi tanımlar

## KULLANIM

```
CREATE AGGREGATE isim (
    BASETYPE = girdi_veri_türü,
    SFUNC = durum_işlevi,
    STYPE = durum_veri_türü
    [ , FINALFUNC = son_işlev ]
    [ , INITCOND = ilk_değer ]
)
```

## Açıklama

**CREATE AGGREGATE** yeni bir ortak değer işlevi tanımlar. Temel ve çok kullanılan ortak değer işlevleri dağıtıma dahil edilmiş ve bunlar

<http://www.postgresql.org/docs/8.0/static/functions-aggregate.html>

adresinde açıklanmıştır. Eğer yeni türlerin tanımlanması ya da dağıtımla gelmeyen bir ortak değer işlevine ihtiyaç olursa istenen özellikleri sağlamak için **CREATE AGGREGATE** kullanılabilir.

Bir şema ismi verilmişse (**CREATE AGGREGATE myschema.myagg . . .** gibi), ortak değer işlevi belirtilen şema içinde oluşturulur.

Bir ortak değer işlevi, ismi ve girdi veri türü ile tanımlanır. Eğer iki ortak değer işlevi farklı veri türleri ile çalışıyorlarsa, aynı şema içinde aynı isimle tanımlanabilir. Ayrıca, bir ortak değer işlevi gerek ismi gerekse girdi veri türü bakımından aynı şema içindeki sıradan işlevlerden de farklı olmalıdır.

Bir ortak değer işlevi bir ya da iki sıradan işlevden oluşur: *durum\_işlevi* ile belirtilen bir durum geçiş işlevi ve *son\_işlev* ile isteğe bağlı olarak belirtilebilen bir son hesap işlevi. Bunlar şöyle kullanılır:

```
durum_işlevi( dahili_durum, sonraki-veri-ögesi )
    ---> sonraki-dahili-durum
son_işlev( dahili_durum )
    ---> ortak-değer
```

PostgreSQL ortak değerlerin o anki dahili durumunu saklamak üzere *durum\_veri\_türü* ile belirtilen veri türünde bir geçici değişken oluşturur. Her girdi ögesinde, yeni bir dahili durum değerini hesaplamak üzere durum geçiş işlevi çağrılır. Tüm veri işlendikten sonra, döndürülecek ortak değeri hesaplamak için son işlev çağrılır. Eğer bir son işlev yoksa, son durum değeri olduğu gibi döndürülür.

Bir ortak değer işlevine dahili durum değeri için bir ilk değer olarak, bir *ilk\_değer* sağlanabilir. Bu veritabanında `text` türünde bir sütun olarak belirtilir ve saklanır, fakat bu, durum değerinin veri türündeki bir sabitin geçerli bir harici gösterimi olmalıdır. Eğer verilmezse, durum değeri NULL olarak ayrılır.

Eğer durum geçiş işlevi argümanlarına bağımlı olarak bildirilmişse, NULL girdilerle çağrılmaz. Böyle bir geçiş işleviyle cümle çalıştırıldığında davranışı şöyle açıklayabiliriz. NULL girdi değerleri yoksayılar (işlev çağrılmamıştır ve önceki durum değeri beklemektedir). Eğer ilk durum değeri NULL ise, ilk NULL olmayan girdi değeri durum değeri yapılır ve geçiş işlevi ikinci NULL olmayan girdi değeriyle çağrılır. Bu, `max` gibi ortak değer işlevlerini gerçeklerken kullanışlıdır. Bu davranış sadece *durum\_veri\_türü* ile *girdi\_veri\_türü* aynı ise kullanılabilir. Bu veri türleri farklı olduğunda, ya NULL olmayan bir *ilk\_değer* ya da argümanlarına bağımlı olmayan bir geçiş işlevi sağlamanız.

Eğer durum geçiş işlevi argümanlarına bağımlı değilse, koşulsuz olarak her girdi değeriyle çağrılacak, NULL girdi değerleriyle ve kendinin NULL geçiş değerleriyle çalışacaktır. Bu, işlev yazarının, ortak değerlerin boş değerlerle elde edilebilmesi bakımından tam denetime sahip olmasıyla mümkün olur.

Eğer son işlev argümanlarına bağımlı olarak bildirilmişse, son durum değeri NULL olduğunda çağrılmayacak; bunun yerine kendiliğinden bir NULL sonuç döndürülecektir. (Şüphesiz bu, argümanlarına bağımlı olarak işlem yapabilen işlevler için normal bir davranıştır.) Her durumda, bir son işlev bir NULL değer döndürecek bir seçeneğe sahip olabilir. Örneğin, `avg` işlevinin son işlevi sıfır girdi satırı varsa, NULL döndürür.

## Parametreler

### *isim*

Oluşturulacak ortak değer işlevinin ismi (şema nitelemeli olabilir).

### *girdi\_veri\_türü*

Bu ortak değer işlevinin üzerinde işlem yapacağı girdinin veri türü. Girdi değerlerinin türüyle ilgilenmeyen bir ortak değer işlevi için (`count (*)` gibi), "ANY" olarak belirtilebilir.

### *durum\_işlevi*

Her girdi değeri için çağrılacak durum geçiş işlevinin ismi. Bu normalde iki argümanlı bir işlevdir; ilk argümanı *durum\_veri\_türü* türünde, ikincisi ise *girdi\_veri\_türü* türündedir. Eğer bir ortak değer işlevi girdi veri türü ile ilgilenmiyorsa işlev *durum\_veri\_türü* türünde tek bir argüman alabilir. Her durumda işlev *durum\_veri\_türü* türünde bir değer döndürmelidir. Bu işlev o anki durum değerini ve o anki girdi veri ögesini alıp, sonraki durum değeri ile döner.

### *durum\_veri\_türü*

Durum geçiş işlevinin durum değerinin veri türü.

### *son\_işlev*

Tüm girdi değerleri işleme sokulup, ortak değer hesaplanacağı zaman çağrılan son işlevin ismi. İşlev *durum\_veri\_türü* türünde tek bir argüman almalıdır. Ortak değer işlevinin dönüş türü bu işlevin dönüş türü olarak tanımlanır. Eğer *son\_işlev* belirtilmemişse, son durum değeri işlevin dönüş değeri olarak kullanılır ve dönüş türü *durum\_veri\_türü* olur.

### *ilk\_değer*

Durum değeri için ilk değer. *durum\_veri\_türü* türünde kabul edilebilecek bir dizge sabit olmalıdır. Belirtilmezse, durum değeri NULL olarak ayrılır.

**CREATE AGGREGATE** deyiminin parametrelerini tam da yukarıdaki sırasıyla yazmak şart değildir, istenen sırada yazılabilir.

## Örnekler

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

## Uyumluluk

**CREATE AGGREGATE** bir PostgreSQL oluşumdur. SQL standardı kullanıcı tanımlı ortak değer işlevlerini desteklememektedir.

## İlgili Belgeler

**ALTER AGGREGATE** [`alter_aggregate(7)`],  
**DROP AGGREGATE** [`drop_aggregate(7)`].

## Çeviren

Nilgün Belma Bugüner <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 AGGREGATE(7)

Bu dosya (man7-create\_aggregate.pdf), belgenin XML biçiminin T<sub>E</sub>XLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

31 Ocak 2007