

İsim

ALTER TABLE – bir tablonun tanımını değiştirir

KULLANIM

```
ALTER TABLE [ ONLY ] isim [ * ]
    eylem [, ... ]
ALTER TABLE [ ONLY ] isim [ * ]
    RENAME [ COLUMN ] sütun TO yeni_sütun
ALTER TABLE isim
    RENAME TO yeni_isim
```

eylem şunlardan biri olabilir:

```
ADD [ COLUMN ] sütun tür [ sütun_kısıtı [ ... ] ]
DROP [ COLUMN ] sütun [ RESTRICT | CASCADE ]
ALTER [ COLUMN ] sütun TYPE tür [ USING ifade ]
ALTER [ COLUMN ] sütun SET DEFAULT ifade
ALTER [ COLUMN ] sütun DROP DEFAULT
ALTER [ COLUMN ] sütun { SET | DROP } NOT NULL
ALTER [ COLUMN ] sütun SET STATISTICS tamsayı
ALTER [ COLUMN ] sütun
    SET STORAGE { PLAIN | EXTERNAL | EXTENDED | MAIN }
ADD tablo_kısıtı
DROP CONSTRAINT kısıt_ismi [ RESTRICT | CASCADE ]
CLUSTER ON indeks_ismi
SET WITHOUT CLUSTER
SET WITHOUT OIDS
OWNER TO yeni_aidiyet
SET TABLESPACE tabloalanı_ismi
```

Açıklama

ALTER TABLE mevcut bir tablonun tanımını değiştirir. Çeşitli alt biçimleri vardır:

ADD COLUMN

CREATE TABLE [*create_table*(7)] ile aynı sözdizimini kullanarak tabloya yeni bir sütun eklemekte kullanılır.

DROP COLUMN

Bir tablodan bir sütunu kaldırmak için kullanılır. Sütunla ilgili indeksler ve tablo kısıtları özdevimli olarak kaldırılır. Anahtar başvuruları ve sanal tablolar gibi sütuna tablo dışında bağımlı herşeyin buna dahil olmasını istiyorsanız, **CASCADE**'a da ihtiyacınız var demektir.

ALTER COLUMN TYPE

Bir tablo sütununun türünü değiştirmekte kullanılır. Sütunla ilgili indeksler ve tablo kısıtları yeni sütun türünü kullanacak şekilde, özgün olarak sağlanmış ifadelerle yeniden çözümlenerek özdevimli olarak değiştirilir. İsteğe bağlı **USING** deyimile yeni sütun değerlerinin eski değerlerden nasıl hesaplanacağı belirtilir; kullanılmadığı takdirde, ötanımlı dönüşüm, eski veri türünden yeni veri türüne atama dönüşümü şeklinde yapılır. Eski türden yeni türe dönüşüm için bir örtük dönüşüm ya da atama dönüşümü yoksa, bir **USING** deyimini kullanılmak zorundadır.

SET/DROP DEFAULT

Bir sütunun öntanımlı değerini belirlemek ya da kaldırmak için kullanılır. Öntanımlı değerler, sadece sonraki **INSERT** cümlelerine uygulanır; tablodaki mevcut satırları etkilemez. Öntanımlılar, ayrıca sanal tablolar için de oluşturulmuş olabilir, bu durumda sadece, sanal tabloya **ON INSERT** uygulanmadan önce **INSERT** deyimiyle yerleştirilir.

SET/DROP NOT NULL

Bir sütunun NULL değerlere izin verip vermeyeceğini belirlemek için kullanılır. Sadece, sütun bir NULL değer içermediği takdirde, **SET NOT NULL** kullanabilirsiniz.

SET STATISTICS

Sonraki **ANALYZE** [`analyze (7)`] işlemleri için sütun başına istatistik toplama hedefini belirlemek için kullanılır. Hedef 0–1000 aralığında ya da sistem öntanımlı istatistik hedefininin (bir yapılandırma parametresi olan `default_statistics_target`'in değerinin) kullanılması için –1 olarak belirtilebilir. PostgreSQL sorgu planlamacısı tarafından istatistiklerin kullanımı hakkında daha fazla bilgi edinmek için

<http://www.postgresql.org/docs/8.0/static/planner-«stats.html> adresine bakınız.

SET STORAGE

Bir sütunun saklama kipini belirlemek için kullanılır. Bu, sütunun içerde mi yoksa ek bir tabloda mı tutulacağını ve verinin sıkıştırılacağını mı yoksa sıkıştırılmayacağını mı denetler. `integer` gibi sabit genişlikli değerlerin için içerde ve sıkıştırılmış olması için **PLAIN** kullanılmalıdır. İçerde ve sıkıştırılmış değerler için **MAIN**, harici ve sıkıştırılmamış değerler için **EXTERNAL**, harici ve sıkıştırılmış değerler için **EXTENDED** kullanılır. **EXTENDED**, **PLAIN** kipini desteklemeyen çoğu veri türü için öntanımlıdır. **EXTERNAL** kullanımı `text` ve `bytea` türündeki sütunlar için saklama alanını büyütmesine karşın altdizge işlemlerinin, daha hızlı yapılmasını sağlar. **SET STORAGE** tablo üzerindeki hiçbir şeyi kendisi değiştirmez, sadece gelecekteki tablo güncellemeleri sırasında izlenecek stratejiyi belirler. Daha fazla bilgi için

<http://www.postgresql.org/docs/8.0/static/storage-«toast.html> adresine bakınız.

ADD *tablo_kısıtı*

CREATE TABLE [`create_table (7)`] ile aynı sözdizimini kullanarak tabloya yeni bir kısıt eklemekte kullanılır.

DROP CONSTRAINT

Bir tablodaki kısıtları kaldırmak için kullanılır. Şimdilik, tablolardaki kısıtların eşsiz isimlere sahip olması gerekmektedir, dolayısıyla belirtilen isimle eşleşen birden fazla kısıt olabilir. Eşleşen tüm kısıtlar kaldırılacaktır.

CLUSTER

Sonraki **CLUSTER** [`cluster (7)`] işlemleri için öntanımlı indeksi seçmekte kullanılır. Gerçekte tabloyu yeniden kümelemez.

SET WITHOUT CLUSTER

Tabloda en son kullanılan **CLUSTER** [`cluster (7)`] indeks belirtimini kaldırmak için kullanılır. Bu, gelecekteki bir indeks belirtmeyen tüm kümeleme işlemlerini etkileyecektir.

SET WITHOUT OIDS

Tablodan `oid` sistem sütununu kaldırmak için kullanılır. Zaten bir `oid` sütununun bulunmadığından şikayet etmesi dışında, **DROP COLUMN oid RESTRICT** deyimine tamamen eşdeğerdir.

Kaldırılmış bir `oid` sütununun tekrar yerine konulmasını sağlayacak bir **ALTER TABLE** alt biçimi olmadığına dikkat edin.

OWNER

Tablonun, kayıt listesinin veya sanal tablonun sahibini belirtilen kullanıcı yapar.

SET TABLESPACE

Tablonun tablo alanını değiştirmek için kullanılır. Tablo ile ilişkili tüm veri dosyalarının yani tablo alanına taşınmasını sağlar. İndeksler eğer varsa yani alana taşınmaz; taşınması isteniyorsa, ayrı bir cümle olarak bir **SET TABLESPACE** cümlesi ile yapılabilir. Ayrıca **CREATE TABLESPACE** [`create_tablespace(7)`] kılavuz sayfasına de bakınız.

RENAME

Bir tablo (veya indeks, kayıt listesi ya da sanal tablo) ismini ya da bir tablonun tek bir sütununun ismini değiştirmekte kullanılır. Saklanmış veriler etkilenmez.

RENAME dışında tüm eylemler çeşitli değişikliklerin birlikte uygulanması için aynı cümlede kullanılabilir. Örneğin, bazı sütunların isimleri değiştirilirken, bazı sütunların eklenmesi mümkündür. Bu işlem genellikle, tüm değişikliklerin tek bir geçişte yapılması zorunlu dev tablolarda kullanışlıdır.

Çalıştırılması için ayrıcalıklı kullanıcı olmanın gerekli olduğu **ALTER TABLE OWNER** dışında, **ALTER TABLE** kullanabilmek için tablonun sahibi olmalısınız.

Parametreler*isim*

Değiştirilecek tablonun ismi (şema nitelermeli olabilir). Eğer **ONLY** belirtilmişse sadece bu tablo değiştirilir, belirtilmemişse bu tablo ve onun çocukları olan tablolar (varsa) değiştirilir. Çocuk tablolarında değiştirileceğini belirtmek üzere tablo isminin sonuna bir ***** eklenebilirse de, bu artık öntanımlıdır. (7.1 sürümünden önce, **ONLY** öntanımlı davranıştır. Öntanımlı davranış `sql_inheritance` yapılandırma parametresi ile değiştirilebilir.)

sütun

Yeni ya da mevcut bir sütunun ismi.

yeni_sütun

Yeni ya da mevcut bir sütunun yeni ismi.

yeni_isim

Tablonun yeni ismi.

tür

Mevcut bir sütun için yeni veri türü ya da yeni bir sütunun veri türü.

tablo_kısıtı

Tablo için yeni tablo kısıtı.

kısıt_ismi

Kaldırılacak mevcut bir kısıtın ismi.

CASCADE

Kaldırılan kısıt veya sütuna bağımlı nesnelerin (örneğin, bu sütuna başvuran sanal tabloların) özdevimli olarak kaldırılmasını sağlar.

RESTRICT

Bağımlı nesneleri olan kısıt veya sütunun kaldırılmasını engeller. Bu davranış öntanımlıdır.

indeks_ismi

Tablonun kümeleme için imleyeceği indeks ismi.

yeni_aidiyet

Tablonun yeni sahibinin kullanıcı ismi.

tabloalanı_ismi

Tablonun taşınacağı yeni tablo alanı ismi.

Ek Bilgi

COLUMN anahtar sözcüğü kuru gürültüdür ve atlanabilir.

Bir sütun **ADD COLUMN** ile eklendiğinde, tablodaki mevcut satırların tümü sütunun öntanımlı değeriyle ilklendirilir (bir **DEFAULT** deyimi belirtilmemişse NULL ile).

Bir sütunun boş olmayan bir öntanımlı ile eklenmesi veya mevcut bir sütunun türünün değiştirilmesi tablonun tamamının yeniden yazılmasına sebep olur. Bu çok büyük tablolarda önemli sayılabilecek bir süre alır; ayrıca, geçici olarak büyüklüğünün iki katı kadar bir disk alanı gerektirecektir.

Bir **CHECK** veya **NOT NULL** kısıtının eklenmesi mevcut satırların kısıtı yerine getirip getirmediğinin sınanması için tablonun taranmasını gerektirir.

Tek bir **ALTER TABLE** cümlesinde çok sayıda değişikliğin belirtilebilmesi seçeneğinin sağlanmasının ana sebebi çok sayıda tablonun taranmasını ya da tablonun yeniden yazılmasının gerektiği durumlarda tüm değişikliklerin tek bir geçişte yapılmasını sağlamaktır.

DROP COLUMN sütunu fiziksel olarak silmez, sadece SQL işlemleri için görünmez yapar. Tablo üzerindeki sonraki veri girme ve güncelleme işlemleri sütunda boş değer saklanmasına sebep olur. Bu, bir sütunun kaldırılmasını hızlandırırsa da tablonun diskte kapladığı alan, kaldırılan sütunun kapladığı alan geri alınmadığından hemen düşmeyecektir. Mevcut satırlar güncellendikçe bu alan geri kazanılacaktır ama bu çok uzun zaman alabilir.

Aslında **ALTER TABLE** cümlesinin bir tablonun tümünün yeniden yazılmasını gerektirdiği durumda bir getirisi de vardır, yeniden yazma işlemi tablodaki ölü alanların elenmesini sağlar. Örneğin, kaldırılan bir sütun nedeniyle tabloda tutulan alanı hemen geri kazanmak için en hızlı yol,

```
ALTER TABLE tablo ALTER COLUMN birsütun TYPE birtür;
```

yazmaktır. Burada *birsütun* ile mevcut bir sütunu, *birtür* ile ise o sütunun kendi türünü belirtiyoruz. Aslında tabloda bir değişiklik yapılmıyor ama bu cümle tablonun yeniden yazılmasını sağlıyor, böylece faydasız veri temizlenmiş oluyor.

ALTER TYPE alt cümlesinin **USING** seçeneği aslında satırın eski değerleriyle ilişkili bir ifadeyi belirtebilir; yani, bit yandan diğer sütunlara başvururken birini dönüştürüyor olabilir. Bu, **ALTER TYPE** sözdizimi ile çok genel dönüşümler yapılmasını mümkün kılar. Bu esneklikten dolayı, **USING** ifadesi sütunun öntanımlı değerine (varsa) uygulanmaz; sonuç bir öntanımlı için gerekli olan bir sabit ifade olmayabilir. Bu, eski türden yeni türe örtük ya da atamalı tür dönüşümünün bulunmadığında, **USING** seçeneği kullanıldığında bile **ALTER TYPE** alt cümlesinin dönüşümde başarısız olacağı anlamına gelir. Böyle durumlarda, öntanımlı **DROP DEFAULT** ile kaldırılarak **ALTER TYPE** uygulanır ve bunun ardından **SET DEFAULT** kullanılarak uygun yeni öntanımlı değer eklenir. Sütunla ilgili indekslere ve kısıtlara da benzer şeyler uygulanır.

Eğer bir tablonun bir çocuk tablosu varsa, çocuk tablolarda aynı değişiklikler yapılmaksızın bu tablonun bir sütununun türünün ya da isminin değiştirilmesine ve bir sütun eklenmesine izin verilmez. Yani, **ALTER TABLE ONLY** reddedilir. Bu, bir tablonun çocuklarının daima ana tablonun sütunlarını içermesini sağlamak içindir.

Hem ana tabloya hem de çocuk tablolara uygulanan bir **DROP COLUMN** işlemi ile sadece, çocuk tablonun ilgili sütunu aynı zamanda başka bir tablodan miras alınmamışsa ve sütun bağımsız olarak tanımlanmamışsa kaldırılacaktır. Sadece ana tabloya uygulanan **DROP COLUMN** işlemi (**ALTER TABLE ONLY ... DROP COLUMN** gibi), çocuk tablolardan bu sütunları asla kaldırmayacak, onları miras alınmış değil, bağımsız sütunlar olarak imleyecektir.

Sistem katalog tablosunda herhangi bir değişikliğe izin verilmez.

Geçerli parametrelerin açıklamaları için **CREATE TABLE** [create_table(7)] kılavuz sayfasına bakınız. Kalıtım ile ilgili bilgileri ise

<http://www.postgresql.org/docs/8.0/static/ddl.html> bölümünde bulabilirsiniz.

Örnekler

Bir tabloya **varchar** türünde bir sütun eklemek için:

```
ALTER TABLE distributors ADD COLUMN address varchar(30);
```

Bir tablodan bir sütunu kaldırmak için:

```
ALTER TABLE distributors DROP COLUMN address RESTRICT;
```

Mevcut iki sütunun türünü tek bir işlemle değiştirmek için:

```
ALTER TABLE distributors
  ALTER COLUMN address TYPE varchar(80),
  ALTER COLUMN name TYPE varchar(100);
```

UNIX zaman değerini içeren bir sütunun türünü **USING** seçeneğini kullanarak **timestamp with time zone** (zaman dilimli zaman gösterimi) türüne dönüştürmek için:

```
ALTER TABLE foo
  ALTER COLUMN foo_timestamp TYPE timestamp with time zone
  USING timestamp with time zone
    'epoch' + foo_timestamp * interval '1 second';
```

Mevcut bir sütunun ismini değiştirmek için:

```
ALTER TABLE distributors RENAME COLUMN address TO city;
```

Mevcut bir tablonun ismini değiştirmek için:

```
ALTER TABLE distributors RENAME TO suppliers;
```

Bir sütuna boş değer içermeme kısıtı eklemek için:

```
ALTER TABLE distributors ALTER COLUMN street SET NOT NULL;
```

Bir sütundan boş değer içermeme kısıtını kaldırmak için:

```
ALTER TABLE distributors ALTER COLUMN street DROP NOT NULL;
```

Bir tabloya bir sınıma kısıtı eklemek için:

```
ALTER TABLE distributors
  ADD CONSTRAINT zipchk CHECK (char_length(zipcode) = 5);
```

Bir tablodan ve çocuklarından bir sınıma kısıtını kaldırmak için:

```
ALTER TABLE distributors DROP CONSTRAINT zipchk;
```

Bir tabloya bir anahtar kısıtı eklemek için:

```
ALTER TABLE distributors
ADD CONSTRAINT distfk FOREIGN KEY (address)
REFERENCES addresses (address) MATCH FULL;
```

Bir tabloya (çoksütunlu) bir eşsizlik kısıtı eklemek için:

```
ALTER TABLE distributors
ADD CONSTRAINT dist_id_zipcode_key UNIQUE (dist_id, zipcode);
```

Bir tabloya bir birincil anahtarı olmasa bile, özdevimli isimlenen bir birincil anahtar kısıtı eklemek için:

```
ALTER TABLE distributors ADD PRIMARY KEY (dist_id);
```

Bir tabloyu başka bir tablo alanına taşımak için:

```
ALTER TABLE distributors SET TABLESPACE fasttablespace;
```

Uyumluluk

ADD, **DROP** ve **SET DEFAULT** alt biçimleri SQL standardıyla uyumludur. Diğer alt biçimler birer PostgreSQL oluşumdur. Ayrıca, tek bir **ALTER TABLE** cümlesinde birden fazla değişikliğin belirtilebilmesi özelliği de bir PostgreSQL oluşumdur.

ALTER TABLE DROP COLUMN ile geriye bir sıfır sütunlu tablo kalmak üzere bir tablonun tek sütunu kaldırılabilir. SQL standardı sıfır sütunluk tablolara izin vermediğinden bu da bir PostgreSQL oluşumdur.

Çeviren

Nilgün Belma Bugüner <[nilgun \(at\) belgeler.gen.tr](mailto:nilgun (at) belgeler.gen.tr)>, Mart 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

ALTER TABLE(7)

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

31 Ocak 2007