

İsim

CREATE INDEX – yeni bir indeks tanımlar

KULLANIM

```
CREATE [ UNIQUE ] INDEX isim ON tablo [ USING yöntem ]
    ( { sütun | ( ifade ) } [ işleç_sınıfı ] [, ...] )
    [ TABLESPACE tablo_alanı ]
    [ WHERE dayanak ]
```

Açıklama

CREATE INDEX belirtilen bir tablo üzerinden bir indeks oluşturur. İndekslerin birincil kullanım amacı veritabanı başarımını arttırmaktır (ancak, uygunsuz kullanımı başarımın düşmesiyle sonuçlanır).

İndeks için anahtar alanları sütun isimleri olarak ya da parantez içinde yazılmış ifadeler olarak belirtilir. İndeksleme yöntemi çöksütunlu indeksleri destekliyorsa, çok sayıda alan belirtilebilir.

Bir indeks alanı tablonun satırındaki bir veya daha fazla sütun değerinden hesaplanan bir ifade olabilir. Bu özellik, bazı temel veri dönüşümlerini temel alan veriye daha hızlı erişim sağlamak için kullanılabilir. Örneğin, `upper(col)` 'a göre hesaplanan bir indeks, **WHERE upper(col) = 'JIM'** gibi bir deyimini belirtilebilmesini mümkün kılar.

PostgreSQL B-tree, R-tree, hash ve GiST indeksleme yöntemlerini içerir. B-tree indeksleme yöntemi bir Lehman–Yao yüksek bilinirlikli B-tree gerçeklenimidir. R-tree indeksleme yöntemi, Gutman'ın dördül ayırma algoritması kullanılarak gerçekleştirir. Hash indeksleme yöntemi ise Litwin'in bir gerçeklenimidir. Kullanıcılar da kendi indeksleme yöntemini tanımlayabilir ama bu oldukça karmaşıktır.

WHERE deyiminin varlığında, bir kısmî indeks oluşturulur. Bir kısmî indeks, bir tablonun kalanına göre indeksleme bakımından daha kullanışlı olan bir bölümündeki girdileri içeren bir indekstir. Örneğin, hem toptan hem de perakende siparişleri içeren bir tablomuz olsun, perakende siparişlerin tabloda daha az bulunmasına karşın en sık kullanılanların bunlar olduğunu varsayalım. Sadece bu kayıtlarla ilgili bir indeks oluşturarak başarıyı önemli ölçüde arttırabilirsiniz. Başka bir olası kısmî indeks uygulaması, tablonun bir alt kümesindekilerin eşsizliğini sağlamak için **WHERE** ile **UNIQUE**'in birlikte kullanımıdır.

WHERE ile kullanılan ifade sadece indekslenen tablonun sütunlarına başvurabilir ama sadece indekslenen değil, bütün sütunlar kullanılabilir. Şimdilik, **WHERE** deyimini içinde altsorgular ve ortak değer işlevleri de kullanılamamaktadır. Aynı kısıtlamalar ifadelerdeki indeks alanları için de geçerlidir.

Bir indeks tanımında kullanılan bütün işlevler ve işleçler **IMMUTABLE** imli olmalıdır, yani sonuçlar bunların argümanlarına bağımlı olmalı ve argümanları asla yorumlanabilir olmamalıdır (başka bir tablonun içeriği ya da şimdiki zaman kullanılamaz). Bu sınırlama indeksin davranışının iyi tanımlanmış olmasını sağlar. Bir kullanıcı tanımlı işlevi bir indeks ifadesinde ya da **WHERE** deyiminde kullanmak istiyorssanız, işlevi tanımlarken **IMMUTABLE** imli olmasını sağlamayı unutmayın.

Parametreler

UNIQUE

İndeks oluşturulurken ve her veri eklenişinde, tabloda birbirinin aynı değerler bulunmaması için sistemin sınama yapmasını sağlar. Girdilerin yinelenmesine sebep olacak bir veri girme veya güncelleme işleminin yapılmaya çalışılması bir hata üretecektir.

isim

Oluşturulacak indeksin ismi; burada şema nitelemeli isimler kullanılamaz. İndeks daima tabloyu içeren şemada oluşturulur.

tablo

İndekslencek tablonun ismi (şema nitelemeli olabilir).

yöntem

İndeks için kullanılacak yöntemin ismi. Değer olarak, `btree`, `hash`, `rtree` ve `gist` verilebilir. `btree` öntanımlı yöntemdir.

sütun

Tablo sütunun ismi.

ifade

Tablonun bir ya da daha fazla sütunu ile ilintili bir ifade. İfade yukarıdaki sözdiziminde gösterildiği gibi parantez içinde yazılmalıdır. Ancak, ifade bir işlem çağırısı biçimindeyse parantez içine alınmayabilir.

işleç_sınıfı

Bir işlem sınıfının ismi. Ayrıntılar için aşağıya bakınız.

tablo_alanı

İndeksin oluşturulacağı tablo alanı. Belirtilmezse, `default_tablespace` yapılandırma değişkeninin değeri; bu değişkene bir değer atanmamışsa veritabanının öntanımlı tablo alanı kullanılır.

dayanak

Bir kısmî indeks için kısıt ifadesi.

Ek Bilgi

İndekslerin hangi durumlarda yararlı olduğu, hangi durumlarda kullanılmayacağı, ne zaman kullanıldığı gibi bilgileri

<http://www.postgresql.org/docs/8.0/static/indexes.html> adresinde bulabilirsiniz.

Şimdilik, çok sütunlu indekslerde sadece B-tree ve GiST indeksleme yöntemleri desteklenmektedir. Öntanımlı olarak en fazla 32 alan belirtilebilir. (Bu sınır PostgreSQL kurulum için yapılandırılırken değiştirilebilir.) B-tree şimdilik sadece eşsiz indeksleri desteklemektedir.

Bir işlem sınıfı bir indeksin her sütunu için belirtilebilir. İşlem sınıfı indeks tarafından sütunlar için kullanılacak işlemleri belirler. Örneğin, dört baytlık tamsayılar üzerindeki bir B-tree indeksi `int4_ops` sınıfını kullanır; bu işlem sınıfı dört baytlık tamsayıların karşılaştırma işlevlerini içerir. Uygulamada, sütun türüne göre öntanımlı olan işlem sınıfı genellikle yeterli olur. İşlem sınıflarına sahip olmanın ana esprisi, birden fazla anlamlı sıralaması olabilen bazı veri türlerinin olmasıdır. Örneğin, bir karmaşık sayı veri türü için sıralamayı hem gerçel kısmı hem de mutlak değeri için yapmak isteyebiliriz. Bunu veri türü için iki işlem sınıfı tanımlayarak ve indeks oluştururken bu sınıflardan birini belirterek yapabiliriz. İşlem sınıfları hakkında daha fazla bilgi edinmek için

<http://www.postgresql.org/docs/8.0/static/indexes-opclass.html> ve

<http://www.postgresql.org/uygulamalar/pgsql/xindex.html> adreslerine bakınız.

Bir indeksi kaldırmak için **DROP INDEX** [`drop_index(7)`] kullanılır.

İndeksler öntanımlı olarak **IS NULL** deyimleri için kullanılmazlar. Böyle durumlarda indeksleri kullanmanın en iyi yolu, **IS NULL** dayatmasını kullanan bir kısmî indeks oluşturmaktır.

Örnekler

`films` tablosunun `title` sütunu üzerinde bir B-tree indeksi oluşturmak için:

```
CREATE UNIQUE INDEX title_idx ON films (title);
```

`films` tablosunun `code` sütunu üzerinde bir indeks oluşturup, bu indeksin `indexspace` tablo alanında kalması için:

```
CREATE INDEX code_idx ON films (code) TABLESPACE indexspace;
```

Uyumluluk

CREATE INDEX bir PostgreSQL oluşumdur. SQL standardında indeksler için bir bahis yoktur.

İlgili Belgeler

ALTER INDEX [`alter_index(7)`], **DROP INDEX** [`drop_index(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 ZİMNEN, 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.