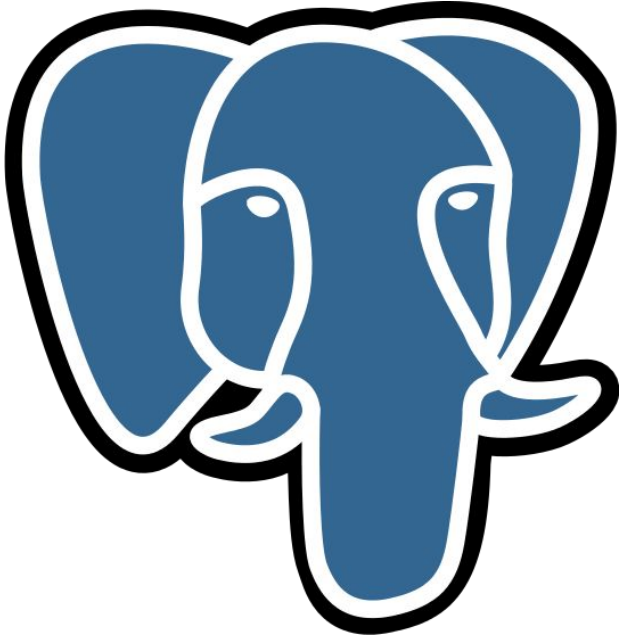


En Güçlü Özellikleri ile PostgreSQL Veritabanı



M.Atıf CEYLAN

@atifceylan



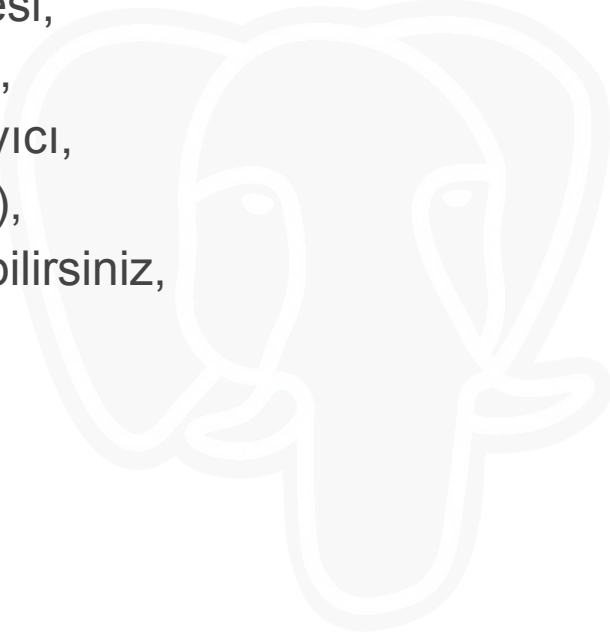
PostgreSQL Nedir?

- Açık kaynak kodlu
- Gerçek cross-platform desteği
- Kurumsal destek
- Hızlı “community” desteği
- <http://postgresql.org>
- <http://postgresql.org.tr>
- Mail listeleri <http://www.postgresql.org/list/>
- pgsql-tr-genel@postgresql.org



Açık Kaynak Kodlu

- 6 Çekirdek takım üyesi,
- 40'tan fazla geliştirici,
- Yüzlerce katkı sağlayıcı,
- Türkiye Grubu (PGD),
- Siz de katkı sağlayabilirsiniz,



En Güçlü Özellikleri

- **Multiple DB on Single Instance**

Çalışan 1 Postgres servisi üzerinden limitsiz veritabanı oluşturabilirsiniz

- **Schema**

Bir DB altında aynı ya da farklı özelliklere sahip tablo, function, view vb. oluşturmaya olanak verir. Aynı bağlantı ve sorgu içerisinde birden fazla şema aynı anda kullanılabilir.

```
SELECT * FROM urun_stok WHERE yil=2015;
```

```
SELECT * FROM arsiv.urun_stok WHERE yil=2011;
```

MVCC

- **Versioning**

Yazma işleminin risksiz (güvenli) yapılmasını sağlar,

- **Transaction**

Veri bütünlüğü/güvenliği ve izolasyon sağlar,

BEGIN;

UPDATE hesaplar SET bakiye = bakiye - 100.00 WHERE isim = 'Atif';

SAVEPOINT savepoint1;

UPDATE hesaplar SET bakiye = bakiye + 100.00 WHERE isim = 'Ali';

ROLLBACK TO savepoint1;

UPDATE hesaplar SET bakiye = bakiye + 100.00 WHERE isim = 'Veli';

COMMIT;

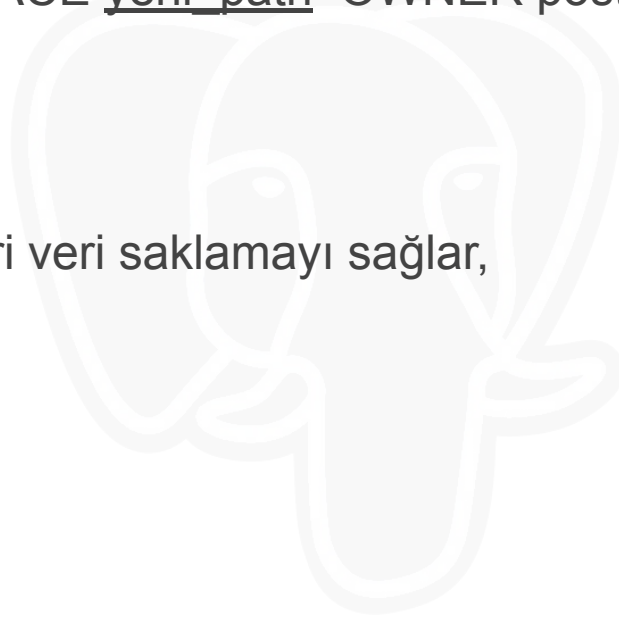
- **Tablespace**

Tablo ve Indekslerin disk(ler) üzerindeki yollarını ayırmaya yarar,

```
CREATE TABLESPACE yeni_path OWNER postgres LOCATION  
'/db/bigdata1';
```

- **Large Objects**

1 kolonda 1GB üzeri veri saklamayı sağlar,



- **Database & Column Level Collation**

Metinlerde dilin gramer yapısına uygun olarak büyütme/küçültme (upper/lower), sıralama (order) yapmayı sağlar. Türkçe'deki küçük "i" büyütüldüğünde "İ" olurken İngilizce'de "I" olur.

```
CREATE DATABASE testdb
```

```
.....  
LC_COLLATE = 'tr_TR.UTF-8'
```

```
LC_CTYPE = 'tr_TR.UTF-8'
```

- **Point-in-Time Recovery (PITR)**

Transaction loglardan en son çalışan zamana dönmek,

- **Replication (Warm, Hot Standby, Streaming Replication)**

- **Kolon bazlı izinler, Kolon bazlı triggers**

Yetkilendirmeleri kolon bazında yapılabilir, tablo satırındaki değişikliği belirli bir kolon değiştiğinde trigger edebilirsiniz.

- **Satır bazlı yetkilendirme**

- **Materialized Views**

Veriler ile birlikte tutulan View'lardır. View özelliğini kullanırken performans artırmayı da sağlar,

- **Triggers**

```
BEGIN
    if new.price > 100 then
        UPDATE order SET discount = discount-20 WHERE date <= current_date;
    end if;
    return old;
END
```

- **Rules**

Trigger gibi her satır etkileşiminden sonra değil, bir sorgu çalıştığında o sorguyu manipüle etmek veya yenisini üretmek için kullanılır. “SELECT” kuralları oluşturulabilir.

- **Inheritance**

Aynı yapısal özelliklere sahip tablolar oluşturmak ve kalıtılmış tablolardaki verilere ana tablo üzerinden müdahale edebilmek için kullanılır,

```
CREATE TABLE my_table
```

```
...
```

```
CREATE TABLE my_table1 (  
  ) INHERITS (my_table)
```

- **Partitioning**

Inheritance özelliği ile çok büyük boyutlu tablolara ait aynı tür veriyi parçalara ayırmak ve bu sayede başarımları elde etmek amacıyla kullanılır,

- **Array, Inet, Cidr, Money, Json, Jsonb vb. Gelişmiş Kolon Tipleri**

array;

```
UPDATE urunler SET ozellikler[0]='Kod007', ozellikler[1]=2.45;  
SELECT ozellikler FROM urunler;
```

Inet;

```
SELECT * FROM my_table WHERE ip >= '10.0.0.15';  
INSERT INTO my_table (ip) VALUES ('10.0.0.255');
```

ERROR: invalid input syntax for type inet: "10.0.0.255"

- **Full Text Search**

Dil bilgisine sahip tam metin arama özelliği. Gelişmiş ve çok hızlı aramalar yapmaya olanak sağlayan bir yapıdır. Özel indexler ve arama ifadeleri kullanılır.

```
INSERT INTO my_table (ad) VALUES( 'Fil Sevenler Derneği' );
```

```
SELECT * FROM my_table WHERE to_tsvector(ad) @@ to_tsquery('Fil&Dernek');
```

- **Window Functions**

Mevcut satırın tablonun diğer satırları ile ilişkili hesaplamalarını yapar.

```
SELECT ucret, sum(ucret) OVER (ORDER BY ucret) FROM personel_maas;
```

ucret	sum
3500	3500
3900	7400
4200	11600
4500	16100
4800	25700

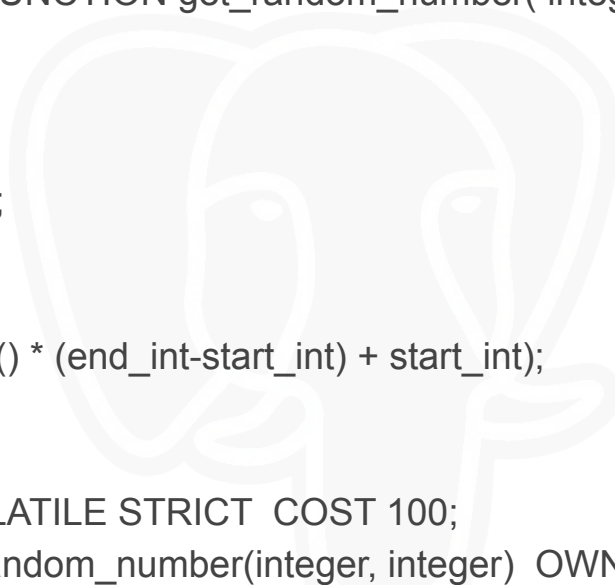
- **CTE (Common Table Expression)**

Çoğu durumda recursive (yinelenebilir) işlemler için fonksiyon yazmaya veya geçici tablolar oluşturmaya ihtiyaç olmaksızın tek SQL ifadesinde yinelenebilir sorgular yazmaya olanak veren sorgulardır.

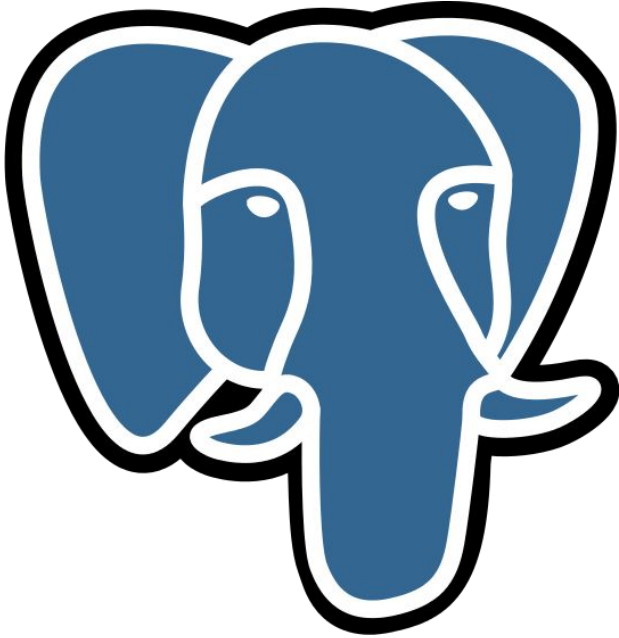
```
WITH RECURSIVE t(n) AS ( VALUES (1) UNION ALL SELECT n+1 FROM t WHERE n < 100 ) SELECT sum(n) FROM t;
```

- **PL/pgSQL, PHP, C, JAVA, PERL, PYTHON ...** Birçok dilde fonksiyon desteği

```
CREATE OR REPLACE FUNCTION get_random_number( integer, integer) RETURNS integer
AS
$BODY$
DECLARE
    start_int ALIAS FOR $1;
    end_int ALIAS FOR $2;
BEGIN
    RETURN trunc(random() * (end_int-start_int) + start_int);
END;
$BODY$
LANGUAGE plpgsql VOLATILE STRICT COST 100;
ALTER FUNCTION get_random_number(integer, integer) OWNER TO postgres;
```



Sorular?



M.Atif CEYLAN

@atifceylan

