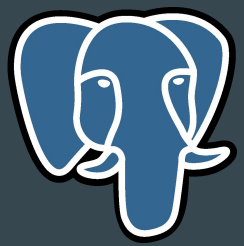
Geliştiriciler için PostgreSQL

...



M.Atıf Ceylan
@atifceylan

AppstoniA

Seminer İçeriği

- ORM Kullanımı
- PostgreSQL Temelleri
- MVCC
- Bağlantı Yönetimi (Connection Pooling)
- JSONB
- Materialized Views
- FTS & Trigram Search
- CTE (Common Table Expression)

Başlarken

AppstoniA

Geliştiricilerin veritabanı efsaneleri.

- NoSQL müthiş(!)
- X bulut sağlayıcının DB servisi var, dehşet! İki tıkla kuruyorsun. Autoscale de oluyor. Proje büyüdüğünde dilediğimiz kadar kaynak büyütebilir ve yükü kaldırabiliriz!
- Nasıl olsa bütün veritabanları aynı. Biri olmazsa diğerine geçerim!
- SQL? Zinhar yazmam! ORM dışına çıkarsam çarpılırım.

PostgreSQL Temelleri

AppsloniA

Kurulum

Linux dağıtımlarının paket yöneticisinde PostgreSQL için temel paketler (dağıtıma veya OS versiyonuna göre de özel paketler) gelmektedir. Sitesinden paket deposunu ekleyip dilediğiniz PostgreSQL sürümünü de yükleyebilirsiniz.

- https://www.postgresql.org/download
- Dağıtım / Sistem seç (Linux, BSD, Windows vb.)
- Paket yöneticisi adımlarını (3-4 basit adım) uygula.

PostgreSQL ve ORM Kullanımı

AppstoniA

ORM Tanımı: (Object-Relational Mapping / Nesne-İlişkisel Eşleme) Veritabanı sorgularının (SQL) nesne tabanlı dillerde nesneler kullanılarak yapılmasını sağlar. Yani SQL -> Nesne (object) eşlemesi yapar.

PostgreSQL ve ORM Kullanımı

```php
<?php
// create_product.php <name>
require_once "bootstrap.php";

$newProductName = 'Kurşun Kalem';

$product = new Product();
$product->setName($newProductName);

$entityManager->persist($product);
$entityManager->flush();

echo "Created Product with ID " . $product->getId() . "\n";
```



```php
<?php
// src/Product.php

use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity
 * @ORM\Table(name="products")
 */
class Product
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue
     */
    protected $id;
    /**
     * @ORM\Column(type="string")
     */
    protected $name;


    public function getId()
    {
        return $this->id;
    }

    public function getName()
    {
        return $this->name;
    }

    public function setName($name)
    {
        $this->name = $name;
    }
```

INSERT INTO products (name)
VALUES ('Kurşun Kalem')

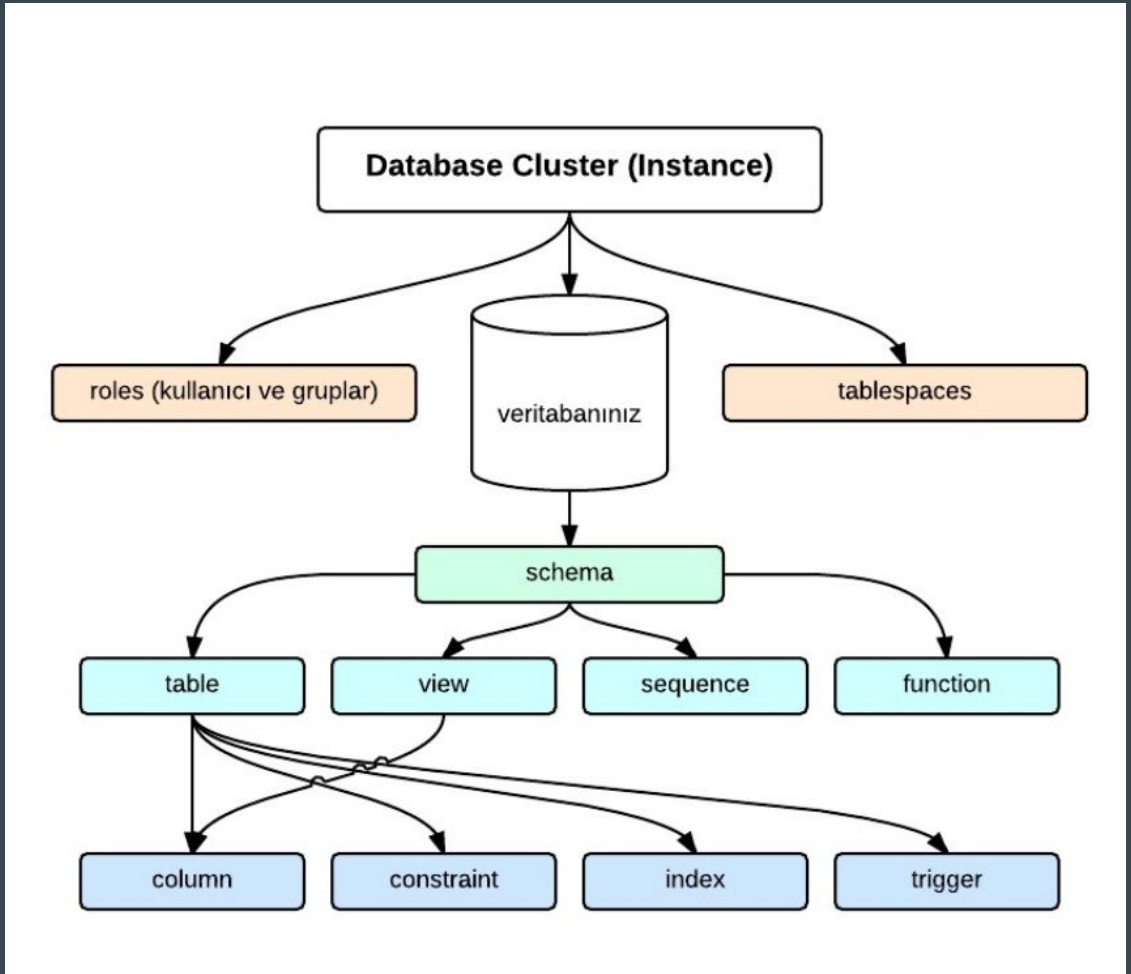
PostgreSQL ve ORM Kullanımı

```php
<?php
// list_products.php
require_once "bootstrap.php";

$productRepository = $entityManager->getRepository('Product');
$products = $productRepository->findAll();

foreach ($products as $product) {
    echo sprintf("-%s\n", $product->getName());
}
```

```php
<?php
// src/Product.php

use Doctrine\ORM\Mapping as ORM;

/**
 * @ORM\Entity
 * @ORM\Table(name="products")
 */
class Product
{
    /**
     * @ORM\Id
     * @ORM\Column(type="integer")
     * @ORM\GeneratedValue
     */
    protected $id;
    /**
     * @ORM\Column(type="string")
     */
    protected $name;


    public function getId()
    {
        return $this->id;
    }

    public function getName()
    {
        return $this->name;
    }

    public function setName($name)
    {
        $this->name = $name;
    }
```

SELECT id, name FROM products

PostgreSQL Temelleri



PostgreSQL Temelleri

AppsloniA

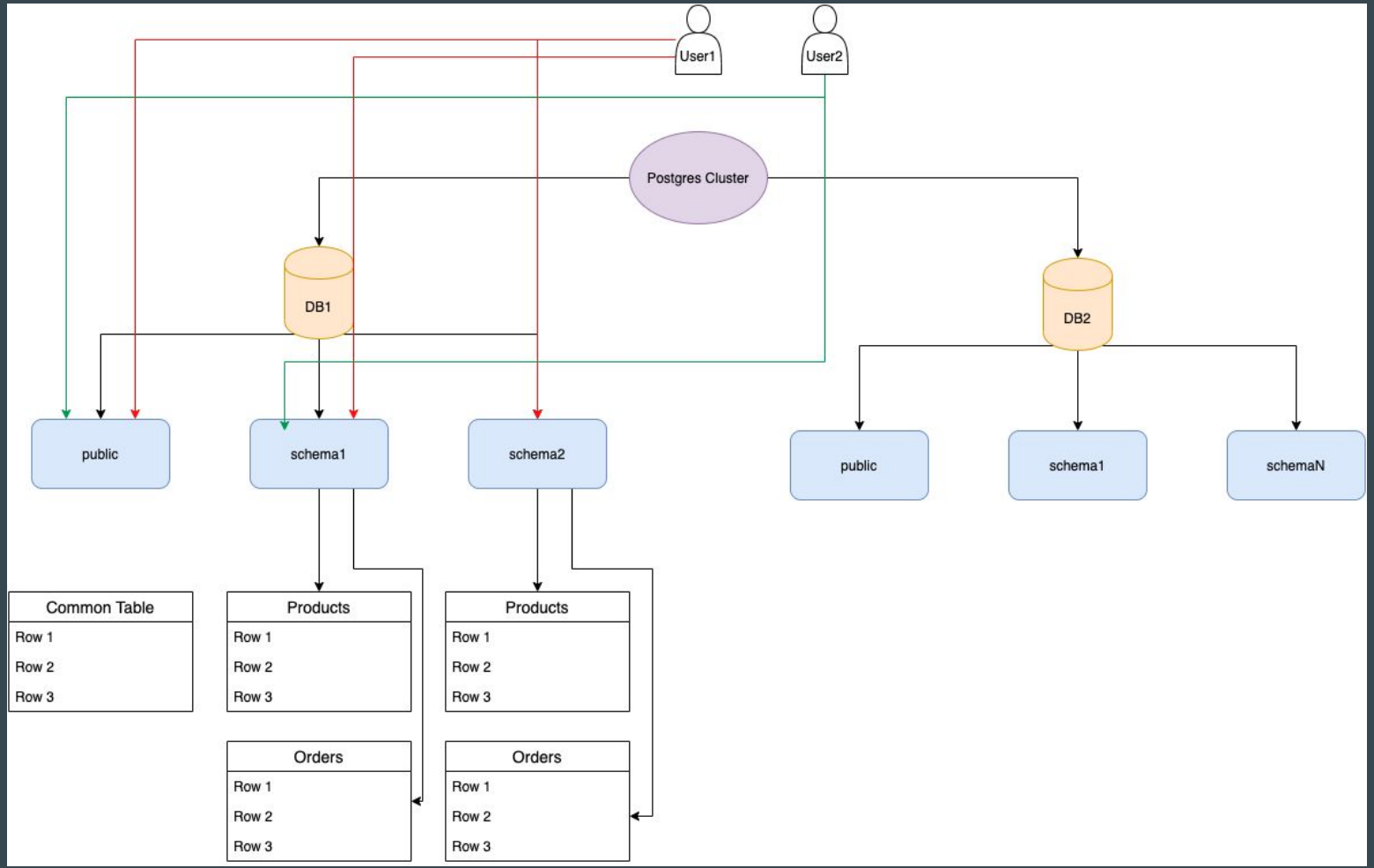
schema

Her bağlantı bir veritabanı seçimi gerektirir. Birden fazla veritabanına aynı anda bağlanılamaz.

Schema;

- Birden çok kullanıcının **mantıksal** olarak ayrılmış yapılara bölünmesine olanak sağlar.
- Aynı bağlantıda birden çok schema'ya erişime izin verir.
- İki farklı schema içerisinde aynı isme sahip birbirinden farklı içeriğe sahip nesneler (tablo, index, view vs.) bulunabilir.
- Aynı bağlantı ve sorgu içerisinde birden fazla şema aynı anda kullanılabilir.

PostgreSQL Temelleri - Schema Erişimi



PostgreSQL Temelleri - schema

AppsloniA

SELECT * FROM products WHERE year=2020;
(public schema altındaki bir tablo)

SELECT * FROM **archive**.products WHERE year=2011;
(archive schema altındaki bir tablo)

PostgreSQL Temelleri - search_path

schema isminin tabloya ön ek olarak verilmeden
doğrudan tablo adı yazılarak sorgulama yapılmasına olanak sağlar.

```
SELECT * FROM archive.products WHERE year=2011;

SET search_path TO archive;

SELECT * FROM products WHERE year=2011;


SHOW search_path;
"$user",public
```

PostgreSQL Temelleri - indexes

AppsloniA

Tanım: Veriye hızlı erişim için kullanılan özel algoritmalar. Eğer index kullanılmıyorsa aranılan veriyi bulmak için bütün veri tek tek kontrol edilir (seq scan).

- Verinin bulunma süresi kısalır.
- İşlemci daha az yorulduğu için toplam performans da artar.
- Özel amaçlı indexler sayesinde (GIN, GIST) yetenekli aramalar (Full Text Search, Trigram Search) veya büyük veri araması / sıralaması (BRIN) yapılabilir.

PostgreSQL Temelleri - indexes

AppsloniA

Index Types

- B-tree (ön tanımlı index type. Hemen her şey için kullanın.)
- BRIN (Çok büyük veriler için)
- GIN, GIST, RUM (FTS & Trigram Search)

PostgreSQL Temelleri - indexes

AppsloniA

İpuçları ve doğru kullanım

- where clause'da kullandığınız alanlarınız indexli olsun.
  (SELECT * FROM products WHERE code='abc123' and category=2)
- Küçük veri setlerinde Postgres index'inizi kullanmayabilir. Panik yok!
- Çok fazla index tanımlamak performans problemlerine neden olabilir. Gereksiz index'e hayır!
- Bir index için en fazla 32 alan aynı anda tanımlanabilir.
- Multi-column index kullanımında bilinçli olun.
- Index oluştururken where clause'da kullanıldığı şekliyle oluşturulmalıdır. Expression index kavramına göz atın.

PostgreSQL Temelleri - indexes

AppstoniA

Expression Index

```
CREATE INDEX idx_customer_email ON customer(email);
```

SELECT * FROM customer WHERE email = 'atifceylan@abc.com';
(idx_customer_email index'i kullanır)

SELECT * FROM customer WHERE email = lower('atifceylan@abc.com');
(idx_customer_email index'i kullanır)

SELECT * FROM customer WHERE lower(email) = 'atifceylan@abc.com';
(idx_customer_email index'i kullanmaz)

SELECT * FROM customer WHERE lower(email) =lower('atifceylan@abc.com');
(idx_customer_email index'i kullanmaz)

```
CREATE INDEX idx_customer_email ON customer( lower(email) );
```

PostgreSQL Temelleri - indexes

AppstoniA

Paritial Index

CREATE INDEX idx_customer_active ON customer(active);

SELECT * FROM customer WHERE active=true and group_id=5;
(idx_customer_active index'i kullanmaz)

CREATE INDEX idx_customer_group ON customer(group_id) **WHERE active = true;**

SELECT * FROM customer WHERE active=true and group_id=5;
(idx_customer_group index'i kullanır)

PostgreSQL Temelleri - indexes

AppstoniA

Index & Order By

- CREATE INDEX idx_info_nulls_low ON customer **(name NULLS FIRST);**
- CREATE INDEX idx_desc_index ON customer **(created_at DESC NULLS LAST);**

MVCC (Multiversion Concurrency Control)

AppsloniA

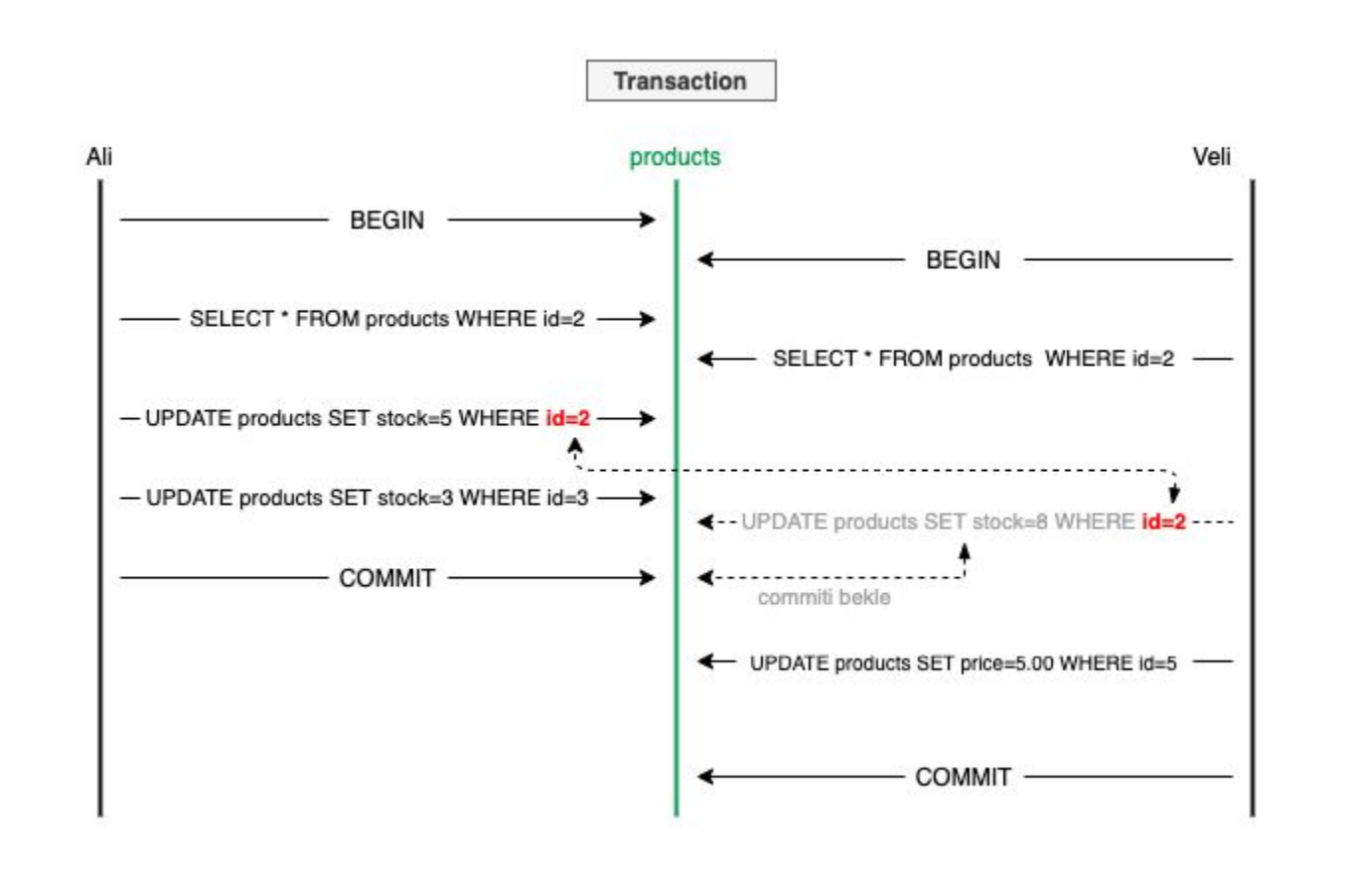
PostgreSQL, eş zamanlı veri erişimlerini yönetmek ve veri tutarlılığını sağlamak için MVCC kullanır. Buna göre her bir SQL statement o an verinin bir versiyonunu görür.

Version; Güncelleme işlemi mevcut satır üzerinde yapılmaz. Mevcut satırın yeni kopyası oluşturularak yapılır. Bu sayede yazmalar okumaları bekletmez.

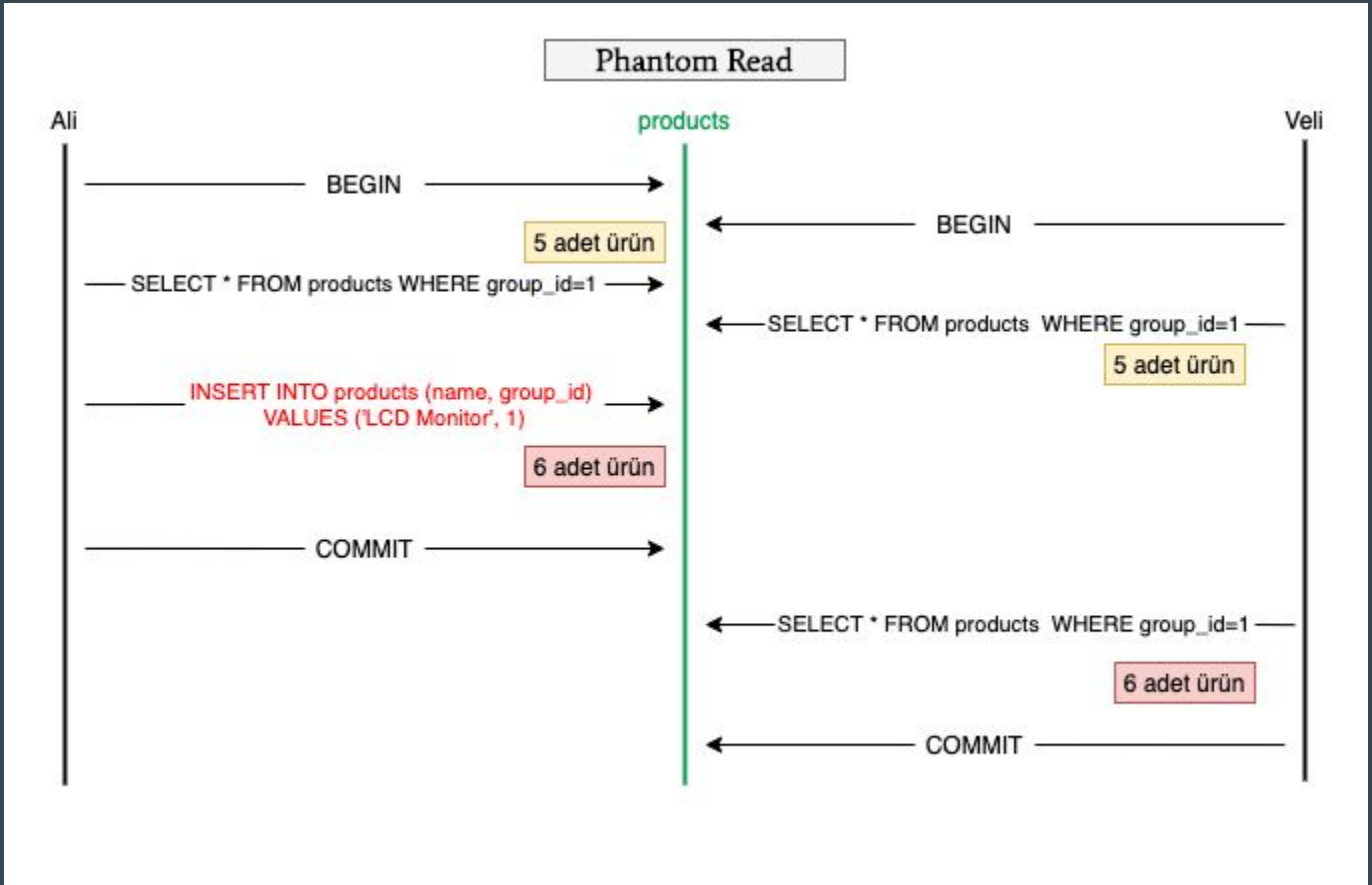
Lock; row veya tablonun yazma ve/veya okuma işlemlerine kapatılmasıdır. Nasıl? Parayla değil sırayla. Yani önce gelen lock'ı kapar!

Bu sürece ben nasıl dahil olurum?

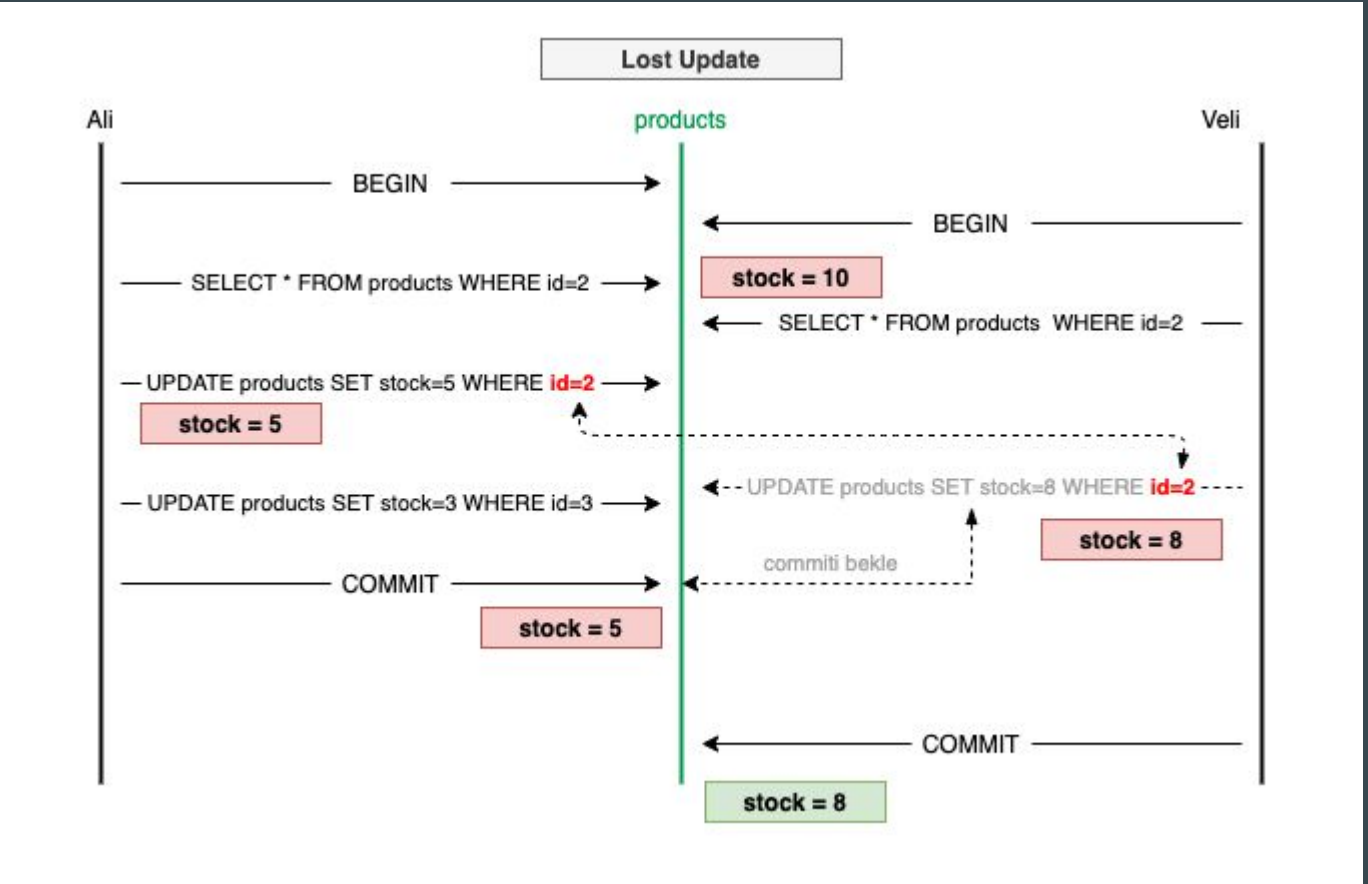
MVCC - Transaction Nedir?



MVCC - Phantom Read



MVCC - Lost Update



MVCC - Transaction Isolation

AppsloniA

Seviyeler

Read Committed (default) default seviye konfigürasyondan değiştirilebilir.

Repeatable Read

Serializable

Anomaliler

Phantom Read

Lost Update

work_mem

AppstoniA

- Önemli bir parametre
- ORDER BY, DISTINCT ve merge join işlemlerinde kullanılan ve kullanıcı bazlı ayarlanabilen bellek parametresi
- Oturum içerisinde birden çok kez farklı değer verilebilir
- Dikkatli kullanmak lazım!

Kullanım:

```
SET work_mem='128MB';
```

JSONB

AppsloniA

- Decomposed binary form olarak tutulur
- Validation desteği
- Elementler indexlenebilir
- Elementler where clause'da belirtilerek arama yapılabilir
- FTS yapılabilir
- Özel fonksiyonlar sayesinde geniş bir kullanım alanına sahip
- İlişkisel modelle birlikte kullanılabilir

JSONB

AppstoniA

```
CREATE TABLE kitap (
        id serial NOT NULL,
        data jsonb
);

INSERT INTO kitap (data) VALUES
('{"title": "Çalıkuşu", "genres": ["Roman", "Aşk"], "published": true}');

INSERT INTO kitap (data) VALUES
('{"title": "Sefiller", "genres": ["Epik", "Kurgusal Tarih "], "authors":["Victor Hugo"], "published":
true}');
```

JSONB

AppsloniA

SELECT data->'title' AS title FROM kitap WHERE **data->'published'** = true;

title

"Çalıkuşu"

"Sefiller"

(2 rows)

CREATE INDEX idx_published ON kitap (**(data->'published')**);

Materialized View

AppsloniA

Normal view kullanımına ek olarak veri saklayabilen view türüdür.

Aşağıdaki örnek, ziyaretçi bazlı sayfa gösterim sayılarını günlük bazda tutmaktadır. 1 kez çalıştırılarak veriler yenilenir. Yenilenene kadar veriler değişmez.

```
CREATE MATERIALIZED VIEW view_count AS
        SELECT date_trunc('day') as day,
                p.visitor_id,
                count(*) as views
        FROM pageview p JOIN visitor v ON (p.visitor_id = v.id)
        GROUP BY date_trunc('day'), p.visitor_id;


REFRESH MATERIALIZED VIEW view_count;
```

FTS (Full Text Search)

AppsloniA

Türkçe de dahil olmak üzere 20'den fazla dili bilen, sözlük bilgisine sahip bir arama modeli.

```
CREATE TABLE films
(
  id          serial      not null,
  name              varchar(255) not null,
  description       text        not null,
  director          varchar(255) not null,
  productor         varchar(255) not null,
  constraint        fts_film_pkey primary key (id)
);
```

	id	name	description	director	productor
1	1	Esaretin Bedeli	Genç ve başarılı bir bankacı olan Andy Du…	Frank Darabont	Niki Marvin, Liz Glotz
2	2	Yüzüklerin Efendisi: Kralın Dönüşü	Aragorn, kendi ırkının çağrısına cevap ve…	Peter Jackson	Peter Jackson, Fran Wa
3	3	Dövüş Kulübü	Oregon Üniversitesinde yüksek lisansını y…	David Fincher, Chon Kye-…	Art Linson, Arnon Milc

FTS (Full Text Search)

AppsloniA

to_tsvector

Sözcükleri birimlerine (lexeme) ayırma

```
SELECT to_tsvector('Bir yolculuk sonrası evinin yanmış olduğunu gördüğünde')

'bir':1 'ev':4 'gördük':7 'olduk':6 'sonras':3 'ya':5 'yolculuk':2


SELECT description, to_tsvector('turkish', description) as document FROM films LIMIT 1;
```

1 row | Comma-...d (CSV) | View Query

	description	document
1	Genç ve başarılı bir bankacı olan Andy Dufresne, karısını ve…	'andy':7 'arkadaş':30 'bankaç':5 'baş':25 'başaracak':46 'başarıl':3 '…

FTS (Full Text Search)

AppsloniA

@@ Operatörü

name: Yüzüklerin Efendisi: Kralın Dönüşü
description: Aragorn, kendi ırkının çağrısına cevap vererek, Orta Dünya'nın bütün kaderi onun elindeyken doğumuyla birlikte ona verilen gücünü kullanabilecek midir? Karanlığın bütün güçleri son savaş için bir araya gelirken Gandalf, Gondor'un yaralı ordusunu toparlamak için hazırlıklara başlar. Gandalf'a gereken destek Rohan Kralı Theoden'den gelir. Thoden, tarihin bu en büyük savaşı için tüm savaşçılarını seferber eder. İçlerinde saklanan Eowyn ve Merry ile birlikte insanlar, tüm cesaretlerine ve ırklarına olan sonsuz bağlılıklarına rağmen Gondor'u kuşatan düşmanların karşısında güçsüzdür. Çok **büyük kayıplar** vereceklerini bilseler de insanlar Sauron'un dikkatini başka yöne çekerek **Yüzük Taşıyıcısı'nın yolculuğunu tamamlamasını** sağlamak için hayatlarının en zor savaşında birbirlerine kenetlenirler.

```
SELECT * FROM films WHERE to_tsvector(description) @@ plainto_tsquery('yüzük taşıyıcısı büyük kayıp yolculuk tamamlamasını');
```

	id	name	description	director	productor
1	2	Yüzüklerin …	Aragorn, kendi ırkının çağrısına cevap vererek, O…	Peter Jackson	Peter Jackson,

Trigram Search

İfadeler veya sözcükler arasındaki benzerliğe göre arama özelliği

```
CREATE EXTENSION pg_trgm;

CREATE INDEX description_trgm_idx ON films USING GIN(description gin_trgm_ops);

SELECT similarity('ifademiz', 'ifademiz')
```

	word_similarity
1	1

```
SELECT word_similarity('ifademiz', 'arama ifademiz içerisinde başka sözcükler de olabilir')
```

	word_similarity
1	1

```
SELECT word_similarity('ifadeniz', 'arama ifademiz içerisinde başka sözcükler de olabilir')
```

	word_similarity
1	0.5555556

Trigram Search

Farklı formlardaki kullanımları

```
SELECT name, description, similarity('hababam', name) as benzer FROM films WHERE name % 'hababam'
```

	name	description	benzer
1	Hababam Sınıfı Tatilde	Okul müdürü, daha fazla para kazanabileceğini düşü…	0.3181818

```
SELECT name, description, similarity('aile', description) as benzer
FROM films
WHERE description %> 'aile' ORDER BY benzer desc
```

	name	description	benzer
1	Tosun Paşa	Seferoğulları ve Tellioğulları aileleri Yeşil Vadi için sürekli didişmek…	0.018050542
2	Dövüş Kulübü	Oregon Üniversitesinde yüksek lisansını yapan Chuck Palanhiuk'un uzak ol…	0.0075642965

```
SELECT name, description, word_similarity('aile', description) as benzer
FROM films WHERE word_similarity('aile',description) > 0.3 ORDER BY benzer desc
```

	name	description	benzer
1	Dövüş Kulübü	Oregon Üniversitesinde yüksek lisansını yapan Chuck Palanhiuk'un uzak ol…	0.8
2	Tosun Paşa	Seferoğulları ve Tellioğulları aileleri Yeşil Vadi için sürekli didişmek…	0.8
3	Yüzüklerin Efend…	Aragorn, kendi ırkının çağrısına cevap vererek, Orta Dünya'nın bütün kad…	0.4

CTE (Common Table Expression)

AppstoniA

Sorguların sonuçlarını aynı sorgu bloğu içinde bir veya daha fazla kez kullanmaya olanak tanır. Bir nevi elde edilen verilerin geçici bir tabloya yazılmış ve oradan okunması gibi düşünülebilir. Fonksiyon yazmak yerine bazı durumlar için CTE kullanılabilir.

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

Result: 5050

Teşekkür

M.Atıf Ceylan

@atifceylan

AppstoniA