

NoSQL Sistemleri

NoSQL Sistemleri Temel Tanımlar
NoSQL Giriş

İçerik

- NSQL kavramını anlama
- İlişkisel vt ile NoSQL sistemleri karşılaştırma
- NoSQL Türleri
- MongoDB CRUD işlemleri
- Hangi proje hangi veri tabanı

Neden NoSQL

- Günümüz yazılım projelerinde
 - veri miktarı,
 - çeşitliliği ve
 - erişim hızı
 - klasik ilişkisel veritabanlarının (RDBMS) sınırlarını zorlamaktadır.
 - Sosyal medya, IoT, büyük veri, mikroservis mimarileri ve gerçek zamanlı uygulamalar; esneklik, yatay ölçeklenebilirlik ve yüksek performans ihtiyacını doğurmuştur.
- Bu ihtiyaçlar NoSQL veritabanlarını ortaya çıkarmıştır.

NoSQL Nedir?

- NoSQL (Not Only SQL), ilişkisel olmayan, şemasız veya esnek şemaya sahip, dağıtık ve ölçülebilir veritabanı sistemlerinin genel adıdır.
- Geleneksel SQL veritabanlarının aksine, NoSQL sistemleri şema esnekliği sunar, yatay ölçülebilirliği destekler ve belirli veri modelleri için optimize edilmiştir.

Temel Özellikler

- Şemasız veya esnek şema
- Yatay ölçeklenebilirlik (scale-out)
- Yüksek okuma/yazma performansı
- Dağıtık mimari
- Büyük veri uyumluluğu
- ACID yerine BASE(Basically Available, Soft state, Eventual consistency)
- Yüksek erişilebilirlik: Kesintisiz hizmet sunma yeteneği

CAP Teoremi

- Consistency
 - Sistemdeki tüm node'lar aynı anda aynı veriyi görür
 - Bir veriyi güncelledin → herkes en güncel halini okur
 - SQL veritabanlarının klasik davranışını buna yakındır
- Availability
 - Sisteme gelen her istek mutlaka bir cevap alır
 - Cevap en güncel olmak zorunda değil
- Partition Tolerance
 - Sunucular arası ağ kopsa bile sistem çalışmaya devam eder
 - Dağıtık sistemlerde kaçınılmazdır
- **Dağıtık (distributed) bir veri tabanı sistemi aynı anda en fazla iki özelliği tam olarak sağlayabilir.**
- **NoSQL sistemleri bu üçlüden ikisini seçer**

İlişkisel Veri Tabanı

- Avantajlar
 - Veri bütünlüğü
 - Karmaşık sorgular için güçlü
 - Finansal ve kritik sistemler için ideal
- Sakıncaları
 - Esnek değil
 - Büyük ölçeklerde performans düşebilir
 - Şema değişikliği maliyetlidir

NoSQL Avantajları

- Esnek şema yapısı - Veri modelleri kolayca değiştirilebilir
- Yatay ölçeklenebilirlik - Trafik artışına karşı kolay ölçeklenebilir
- Yüksek performans - Belirli işlemler için optimize edilmiştir
- Dağıtık mimari - Yüksek erişilebilirlik sunar
- Çeşitli veri modelleri - Farklı ihtiyaçlara uygun seçenekler

NoSQL Sakıncaları

- Tutarlılık sorunları - Eventual consistency nedeniyle veri tutarsızlığı olabilir
- Standart eksikliği - Her NoSQL sistemi farklı API ve dil kullanır
- Transaction sınırlamaları - ACID transaction'ları sınırlıdır
- Olgunluk - Bazı NoSQL sistemleri henüz yeterince olgun değil
- Karmaşık sorgular - JOIN gibi kompleks sorgular zor veya imkansızdır

RDBMS ve NoSQL Karşılaştırması

Özellik	RDBMS - İlişkisel	NoSQL
Veri Yapısı	Tablo	Doküman, Key-Value, Graph
Şema	Katı	Esnek / Şemasız
Ölçekleme	Dikey (scale-up)	Yatay (scale-out)
Performans	Orta	Yüksek
Tutarlılık	Güçlü(ACID)	Nihai Tutarlılık
Kullanım Alanı	Finans, ERP	Sosyal medya, IoT, Big Data
NoSQL, RDBMS'in alternatifi değildir; doğru senaryoda doğru araçtır.		

NoSQL RDBMS Karşılaştırma

Özellik	İlişkisel Vt	NoSQL
SQL Dili	Evet	Çoğunlukla hayır, API tabanlı
ACID Özellikleri	Tam destek	Kısmi destek, eventual consistency
İlişkiler	Join ile güçlü ilişkiler	Genellikle ilişki yok / az
Örnek sistemler	MySQL, MSSQL, PostgreSQL	MongoDB, Cassandra, Redis, Neo4j

Features	Non-Relational	Relational
Availability	High	High
Horizontal Scaling	High	Low
Vertical Scaling	High	High
Data Storage	Optimized for huge data volumes	Medium to large data
Performance	High	Low To Medium
Reliability	Medium	High (Acid)
Complexity	Low	Medium (Joins)
Flexibility	High	Low (Strict-Schema)
Suitability	Suitable For OLAP and OLTP	Suitable For OLTP

Ne Zaman Hangisi Kullanılmalı

NoSQL Tercih Etmek İçin Durumlar:

- Büyük miktarda yapılandırılmamış veya yarı yapılandırılmış veri
- Yüksek ölçeklenebilirlik ihtiyacı
- Hızlı geliştirme süreci (agile development)
- Mikroservis mimarisi
- Gerçek zamanlı büyük veri işleme
- Sosyal medya, IoT, içerik yönetim sistemleri

İlişkisel Veritabanı Tercih Etmek İçin Durumlar:

- Finansal işlemler (ACID gerekliliği)
- Karmaşık sorgular ve raporlama
- Olgun ve standart çözümler
- Veri bütünlüğünün kritik olduğu durumlar
- Sabit ve iyi tanımlanmış veri yapıları

NoSQL Türleri

- Key-Value Veri Tabanları
 - Redis, DynamoDB, Riak
 - Kullanım: Cache, oturum yönetimi, konfigürasyon saklama...
- Document-Based Veri Tabanları
 - MongoDB, CouchDB, Firebase Firestore
 - Kullanım: içerik yönetim sistemleri, katalogolar, kullanıcı profilleri, Json tabanlı uygulamalar...
- Column-Based Veri Tabanları
 - Cassandra, Hbase, Google Bigtable
 - Kullanım: Büyük veri analitiği, log analizi, zaman serileri...
- Graph Veri Tabanları
 - Neo4j, Amazon Neptune, JanusGraph
 - Kullanım: Sosyal ağlar, ilişki analizi, öneri sistemleri...

NoSQL Sistem Türleri Karşılaştırma

	Document Database	Column Store Database	Key-Value Store Database	Graph Database
Performance	High	High	High	Moderate
Availability	High	High	High	High
Flexibility	High	Moderate	High	High
Scalability	High	High	High	Moderate
Complexity	Low	Low	Moderate	High

MongoDB'ye Giriş

- MongoDB, document-based bir NoSQL veritabanıdır.
- Verileri JSON benzeri BSON formatında saklar.

MongoDB Temel Kavramlar

MongoDB	RDBMS Karşılığı
Database	Database
Collection	Table
Document	Row
Field	Column

MongoDB Temel İşlemler(CRUD)

- Database ve Collection
 - use okul
 - db.ogrenciler.insertOne()

Veri Ekleme

```
db.ogrenciler.insertOne({
```

- ad: "Ali",
- soyad: "Yılmaz",
- yas: 22,
- bolum: "Bilgisayar Mühendisliği"
- })

Veri Okuma (Read)

```
db.ogrenciler.find({ bolum: "Bilgisayar  
Mühendisliği" })
```

Veri Silme (Delete)

```
db.ogrenciler.deleteOne({ ad: "Ali" })
```

Şemasız Yapı Örnek

```
db.ogrenciler.insertOne({  
    ad: "Ayşe",  
    dersler: ["Matematik", "Algoritma"],  
    iletisim: {  
        email: "ayse@mail.com",  
        telefon: "5551112233"  
    }  
})
```

- **Aynı collection içinde farklı yapıda dokümanlar olabilir.**

MongoDB Ürün Kataloğu

```
// Ürün koleksiyonu
{
    "_id": ObjectId("61e5a7b8f0123a4567890123"),
    "urun_adi": "Akıllı Telefon",
    "fiyat": 8999.99,
    "stok": 150,
    "kategoriler": ["elektronik", "telefon"],
    "özellikler": {
        "marka": "TechBrand",
        "model": "X200",
        "bellek": "128GB",
        "renk": "Siyah"
    },
    "yorumlar": [
        {
            "kullanici": "Can Demir",
            "puan": 5,
            "yorum": "Harika ürün!"
        }
    ]
}
```

Hangi Senaryoda Hangisi

- İlişkisel veri tabanı
 - Bankacılık
 - Muhasebe
 - Kritik tutarlılık gerektiren sistemler
- NoSQL veri tabanı
 - Sosyal medya
 - Log ve analiz sistemleri
 - IoT
 - Mikroservis mimarileri

Sonuç

- NoSQL yazılım dünyasının önemli bir parçasıdır
- Doğru yerde doğru araç yaklaşımı ile kullanılmalıdır
- İlişkisel veri tabanı ile nosql veri tabanları rakip değildir

Sistem Bileşenleri

1. Öğrenci Bilgi Sistemi

- Öğrenci numarası, ad-soyad, bölüm, sınıf
- Notlar, ders kayıtları
- Akademik geçmiş

2. Etkinlik ve Kulüp Sistemi

- Öğrenci kulüpleri (dinamik sayıda)
- Etkinlikler (konser, seminer, gezi vb.)
- Etkinliğe katılan öğrenciler
- Etkinliğe özel alanlar (bazı etkinliklerde konuşmacı, bazlarında kontenjan, bazlarında online bağlantı linki)

3. Sistem Logları

- Öğrenci giriş-çıkış hareketleri
- Sayfa ziyaretleri
- Hata ve performans kayıtları

