İlişkisel ve İlişkisel Olmayan (NoSQL) Veri Tabanı Sistemleri Mimari Performansının Yönetim Bilişim Sistemleri Kapsamında İncelenmesi

Serdar ÖZTÜRK1, Hatice Ediz ATMACA2

1Yönetim Bilişim Sistemleri, Bilişim Enstitüsü, Gazi Üniversitesi, Ankara,Türkiye

2Endüstri Mühendisliği, Mühendislik Fakültesi, Gazi Üniversitesi, Ankara,Türkiye

GİRİŞ

Belirli bir amaca ulaşmak için elde edilen bilgi organizasyonlar tarafından kısa sürede erişilmek istene faktör haline gelmiştir.
Günümüze kadar olan değişimler verileri modellemede saklanması dolayısıyla veri tabanı kullanılmasını zorunlu hale getirmiştir.
Bu işlemler gerçekleştirilirken ilişkisel veri tabanının yanı sıra ilişkisel olmayan veri tabanı da kullanılmaktadır.
Bu çalışmada iki veri tabanı yönetim sisteminin de mimari performansları kıyaslanmıştır.

BİLİŞİM SİSTEMLERİ VE YÖNETİMİ

Bilişim sistemi, organizasyonlarda bilgi toplamak, düzenlemek ve saklamak olarak tanımlanabilir. Bilgiyi üretmek için 3 aktiviteye ihtiyaç vardır:

- Girdi: Organisazyonun iç ve dış çevresinden bilgi toplamaktır.
- İşlem: Ham veriyi daha anlamlı hale getirir.
- Çıktı: İşlenmiş olan bilgi kullanılacak aktivitelere aktarılır.

Bilişim sistemleri bilişim teknolojisinden yararlanan yönetsel çözümlerdir.

VERİ TABANI VE VERİ TABANI YÖNETİM SİSTEMLERİ

Veri tabanı: Kullanım amacına uygun olarak düzenlenen veri topluluğudur. Veri tabanları nesnelerin birbiriyle olan ilişkisini modeller.

Veri tabanı yönetim sistemleri : Verilerin birbirleriyle bağlantı kurmasını sağlar.

Veri tabanı : Veri tabanı yönetim sistemleri, uygulama programlarını ve kullanıcı ara yüzlerini içeren yapıdır.

Veri tabanı modelleri 8 kategoriye ayrılır.

Tablo modeli : İki boyutlu veri grubundan oluşur. Satırlarda veriler yer alırken sütunlarda ise verilerin benzer özellikleri yer alır.

	Ad Soyad	Kullanıcı Adı	Parola
Kayıt 1	Murat ERGİN	Mergin	kjVdb125
Kayıt 2	Ayşe YILMAZ	Ayılmaz	Bks46db7
Kayıt 3	Can TÜRK	Cturk	fhG8dbt9

Şekil 3.2 Düz veri modeli örneği
(Instance of flat data model)

Hiyerarşik veri modeli: Bu veri tabanın depoladığı verilere kayıt adı verilir. Kayıtlar ağaç mimarisi şeklinde sıralanır.

Ağ veri modeli: Hiyerarşik modelin gelişmiş halidir Bu modelden farkı uç düğüm pozisyonundaki verinin iç düğümü de işaret edebilmesidir.

İlişkisel veri modeli: İlişkiler yardımıyla verinin ilişkileri modellenir.

Nesne yönelimli veri modeli: Nesne yönelimli programlamaya dayanır.

Şekil 3.6 Nesne Yönelimli Veri Modeli
(Object-Oriented Data Model)

Nesne ilişkisel veri modeli : İlişkisel işlevselliğin üzerine nesne yönelimli özellikler içerir. İlk örneği Oracle 8'dir.

Çoklu ortam veri modeli: Nesne ilişkisel veri tabanına benzerdir. Desteklenmesi gereken 3 temel özellik veri miktarı, süreklilik ve senkronizasyondur.

Dağıtık veri modeli: Birden fazla bilgisayar depolanan ve tek ağ üzerinden dağıtılan bilgiler için kullanılır.

VERİ TABANI TASARIMI

Veri tabanı tasarımında gerçek ihtiyaçlar çerçevesinde modellenerek veri tabanına aktarılır.

Şekil 4.1 Veri Tabanı Tasarım Aşamaları

- Kavramsal tasarımda gereksinimlere göre kavramsal şema belirlenir.
- Kavramsal şemada veri tabanının yapısının genel hatları çizilir.
- Fiziksel tasarım aşamasında verinin en yüksek verimi için fiziksel organizasyon sağlanır.
- İç şemada veri tabanının fiziksel olarak gerçekleşmesi için detaylar tanımlanır.

İLİŞKİSEL VE İLİŞKİSEL OLMAYAN VERİ TABANI SİSTEMLERİ

İlişkisel Veri Tabanı

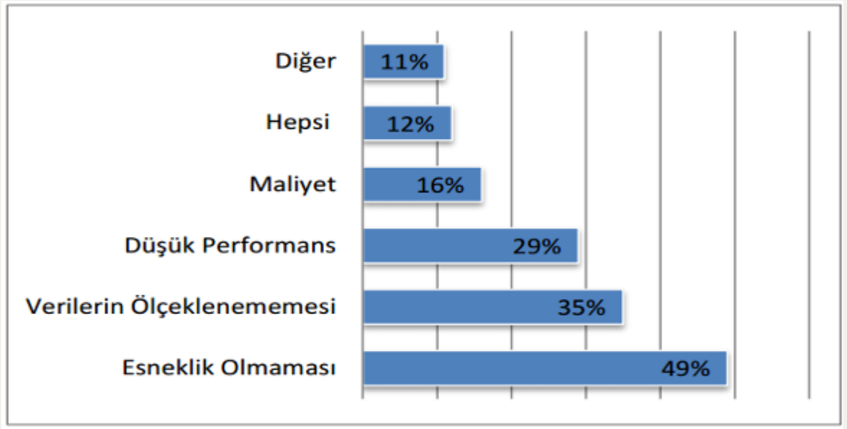
- En yaygın kullanılan veri tabanı sistemlerinden biridir.
- Birbirleriyle ilişkili tablolardan oluşur.
- Her tablo belli yapıdaki verileri saklamak için kullanılır.

İlişkisel veri tabanı sistemlerinde sağlanan 4 temel özellik vardır:

- Bölünmezlik
- Tutarlılık
- İzolasyon
- Dayanıklılık

İlişkisel Olmayan Veri Tabanı

NoSQL, ilişkisel veri tabanı sistemlerine alternatif olarak ortaya çıkmıştır. NoSQL yatay olarak ölçeklendirilen bir veri depolama sistemidir. Araştırmalar sonucunda SQL veri tabanı kullanan kullanıcıların NoSQL veri tabanına geçmek istediği saptanmıştır.



Şekil 5.1 Neden NoSQL Gerekli

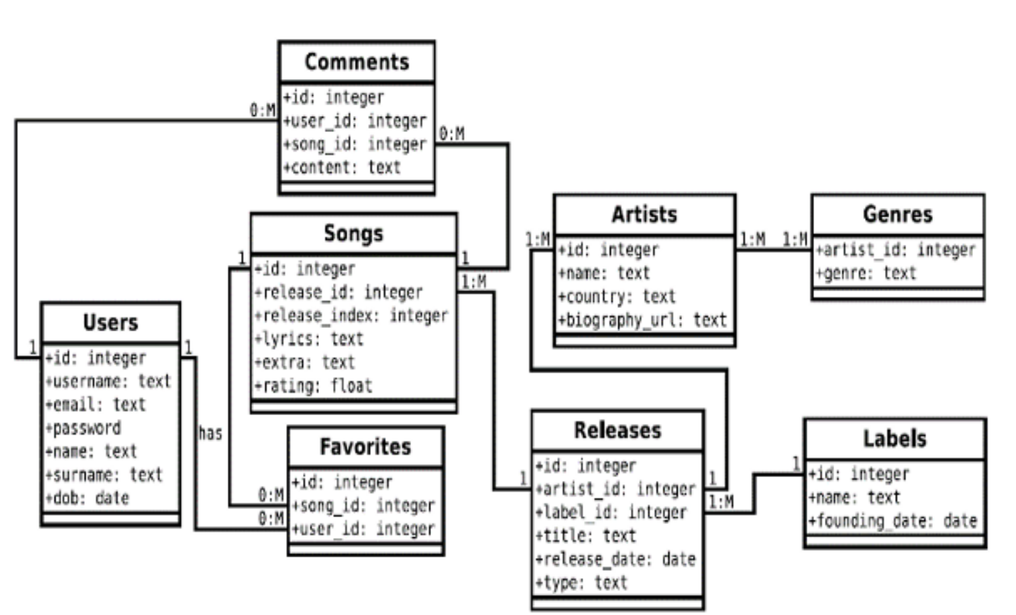
Yandaki tabloda NoSQL veri tabanına geçme nedenleri verilmiştir.
Örneğin Amazon bu gereksinimleri “Dynamo DB”, Google ise “Big Table” adını verdikleri NoSQL veri tabanı sistemi ile çözmektedir.

VERİ TABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRILMASI

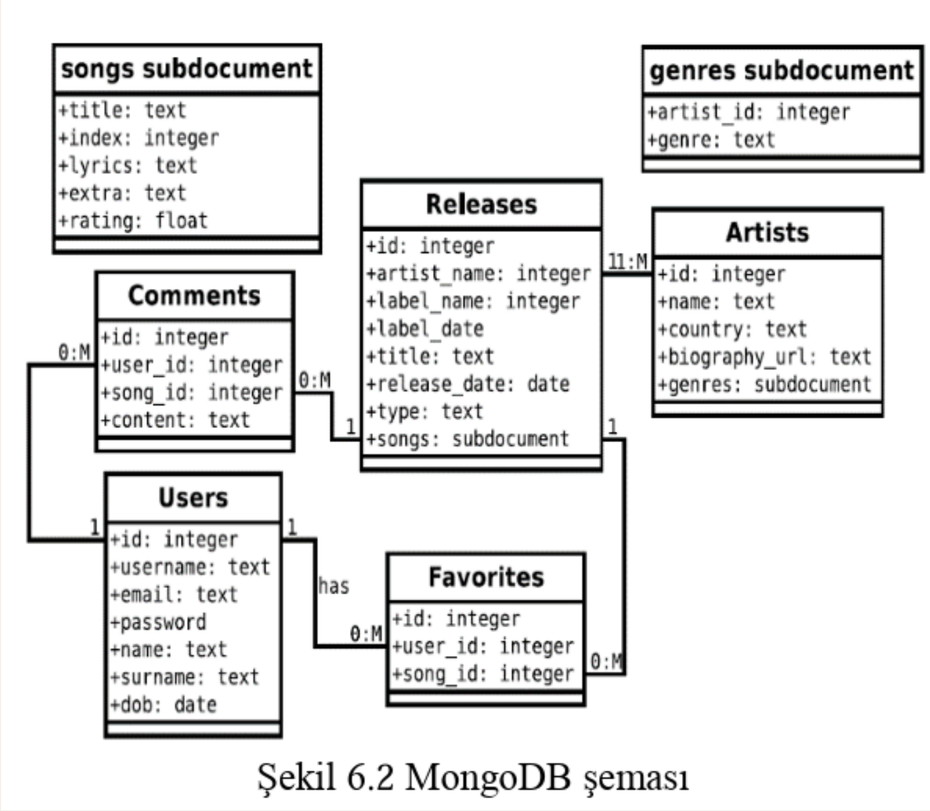
MySQL ve Mongo DB veri tabanı sistemlerinin performans ve yatay ölçeklendirilebilirlik incelemesi için aşağıdaki adımlar uygulanır:

- Veri tabanı sunucu sistemlerinin özelliklerinin belirlenmesi
- Veri tabanı şemalarının oluşturulması
- Veri tabanı ayarlarının yapılması
- Ölçümler ve ölçüm metrikleri bilgileri
- Performans analizi ve sonuçları

Veri tabanı şeması: İki şema tasarlanmıştır.



Şekil 6.1 MySQL veri tabanı şeması



Şekil 6.2 MongoDB şeması

Veri tabanı sorguları: Basitten karmaşığa 3 sorgu kullanılmıştır

Sorgu 1: Basit

```
SELECT * FROM Users WHERE username = 'username '
```

Sorgu 2: Karmaşık

```
SELECT ' Favourites. song_id ' AS fSID , ' Favourites. user_id ' AS fUID
FROM Favourites AS b INNER JOIN Favourites AS a
ON b. user_id = a. user_id
WHERE a. song_id = 123456 AND a. user_i d != 987654
```

Sorgu 3: Detaylı ve karmaşık

```
SELECT ' Songs. release_id ' AS sId , ' Releases. id ' ASrId
FROM Songs INNER JOIN Releases
ON Songs. release_id = Releases. id
WHERE artist_id IN
SELECT ' Genres. artist_id ' AS gAID
FROM Genres AS c
INNER JOIN Artists AS d
ON c.artist_id = d.id WHERE d.name = ' artist_name '
```

Ölçümler: Zaman kavramı ön plana çıkarılmıştır.

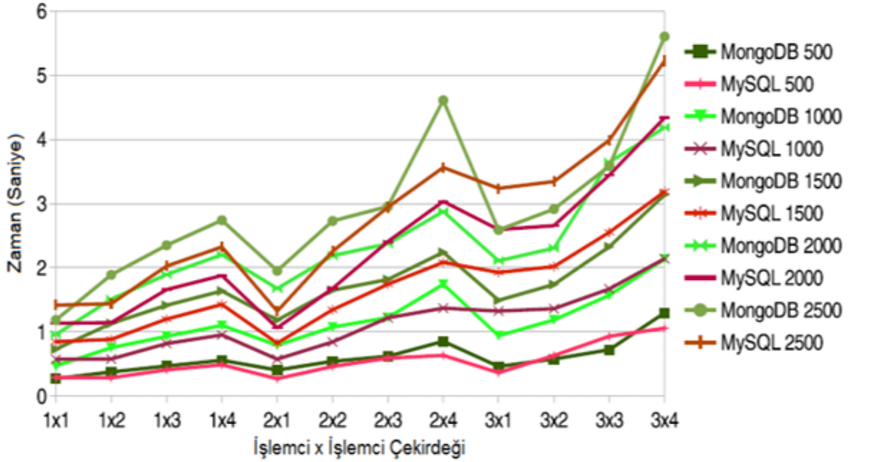
1.Yöntem: Clock () fonksiyonu kullanımı ile belirli bir süre CPU üzerinde harcanan zamanı elde eder.

2. Yöntem: Milisaniye hassasiyetiyle zamanlamarı ayarlayan Gettimeofday () fonksiyonu kullanılır.

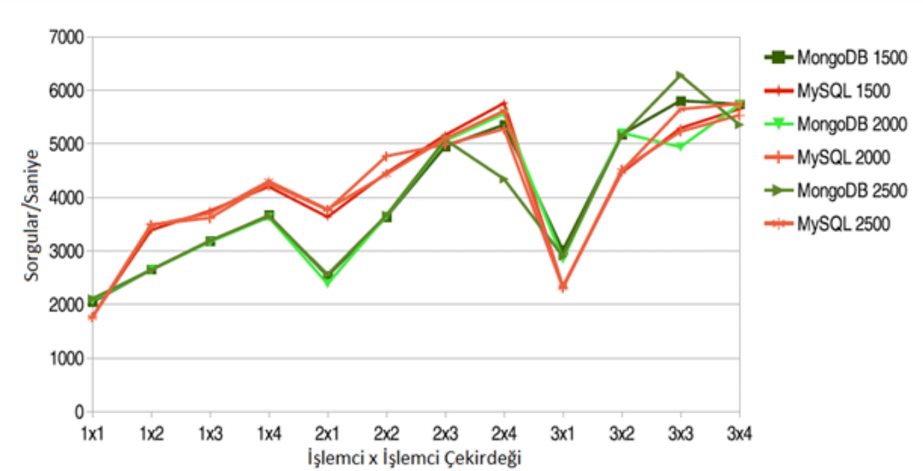
3. Yöntem: Slow Query Log olarak adlandırılır.

Analiz Ve Sonuçlar

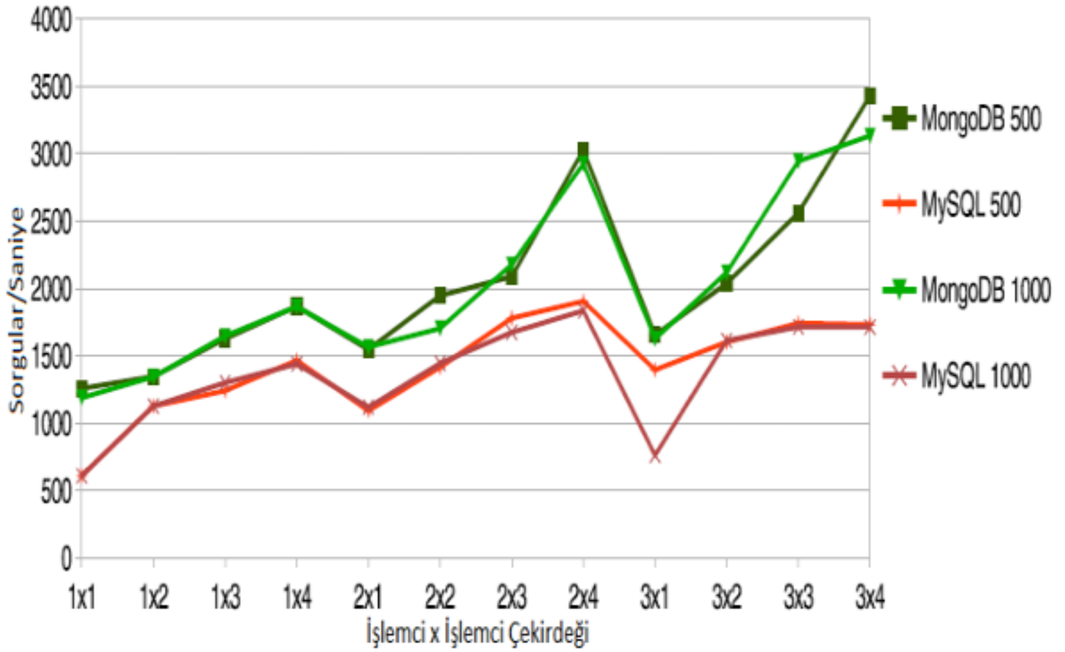
Bu kısımda veri tabanınlarının farklı sorgu türlerine nasıl cevap verdiği ele alınmıştır.Veri tabanı boyutunun performasa etkisi de incelenmiştir. Bu çalışmalarda açıklanan koşullar kapsamında veri tabanlarının kıyaslanabilmesi için çok çeşitli durumlar yaratılmıştır.



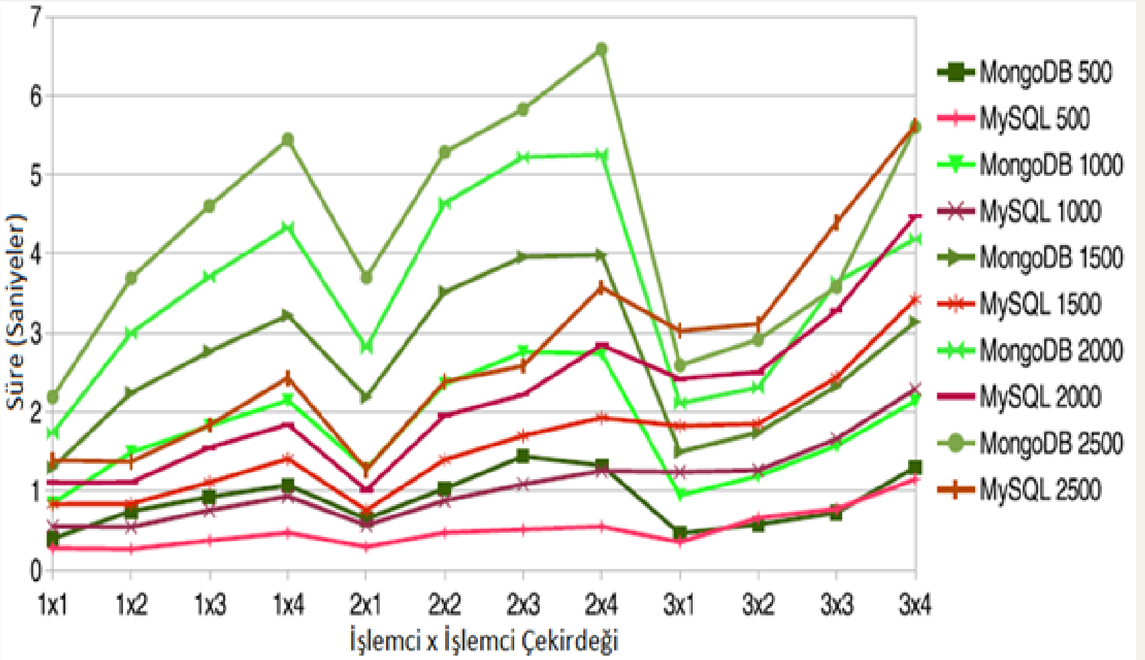
Şekil 6.3 Sorgu 1- Analiz işlemi



Şekil 6.5 Sorgu 1-Çok sayıdaki sorgu miktarı analiz işlemi



Şekil 6.8 Sorgu 2- INNER JOIN ile 500 ve 1000 veri için sorgu/saniye analizi işlemi



Şekil 6.10 Sorgu 3 – Detaylı karmaşık sorgu süre analizi

(Detailed and complex query time analysis)

SONUÇLAR VE DEĞERLENDİRME

Çalışma, ilişkisel ve ilişkisel olmayan (NoSQL) veri tabanı yönetim sistemlerini performans açısından karşılaştırmayı amaçlamaktadır. NoSQL'un esneklik, hız ve ölçeklenebilirlik gibi avantajlarına odaklanırken, ilişkisel veri tabanlarının veri güvenliği ve uygulama taşınabilirliği gibi avantajlarına da değinmektedir. Yapılan testlerde, NoSQL'un büyük veri setlerini daha etkin bir şekilde işleyebildiği ve karmaşık sorgularda daha iyi performans gösterdiği tespit edilmiştir. Ancak, ilişkisel veri tabanları da basit arama sorgularında iyi bir performans sergilemiştir. Sonuç olarak, her iki veri tabanı türünün de avantaj ve dezavantajları bulunmakla birlikte, işletmelerin ihtiyaçlarına ve uygulama gereksinimlerine bağlı olarak en uygun veri tabanı seçiminin yapılması gerektiği vurgulanmıştır.

TEŞEKKÜRLER

YASEMİN FETTAHOĞLU

02210224035