

VERİ ORGANİZASYONU MAKALE ÖZETİ

Giriş: Veri tabanları, veri fazlalığını kontrol ederek ve veri tutarlılığını koruyarak, bilgi yönetimi alanında kritik bir rol oynarlar. Veri entegrasyonu sayesinde, farklı kaynaklardan gelen verilere kolayca erişilebilir, bunlar düzenlenebilir ve paylaşılabilir. Veri depolarının oluşturulması sırasında sıkça kullanılan bir yaklaşım, verilerin uygulama modellerine yakın bir şekilde saklanmasıdır. Bu, verilere erişimin kolaylaşmasını ve veri depolarının etkin kullanılmasını sağlar.

Levent ERGÖREN

02220224046

1-) Hızla ilerleyen bilgisayar ve iletişim teknolojileri, organizasyonları sürekli olarak yeni çözümler bulmaya zorluyor. Bu çözümler, verilerin hızla işlenerek bilgiye dönüştürülmesini gerektiriyor. Bilgi, gün geçtikçe daha hızlı erişilmek istenen en önemli faktör haline gelmiştir. Bilgisayarlar, karar alma süreçlerinde önemli bir rol oynamakta ve "bilgi sistemleri" günümüzün önde gelen konularından biri haline gelmiştir.

Bu değişim ve gelişim, verilerin modellemesini ve saklanması zorunlu kılar. Farklı alanlarda, kurum rehberlerinden işletmelerin bilgilerinin saklanması kadar çeşitli veri modelleme ve depolama ihtiyaçları ortaya çıkar. Verinin büyüklüğü, miktarı ve karmaşıklığı gibi faktörlere bağlı olarak, çeşitli veri modelleme, depolama ve sorgulama yöntemleri geliştirilmiştir.

Veri tabanlarında, ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemleri yoğun olarak kullanılmaktadır. İlişkisel olmayan veri tabanı yönetim sistemleri, performans ve esneklik açısından tercih edilebilir hale gelmiştir.

② 2-) Bilişim sistemi, bir organizasyonda bilgiyi toplama, düzenleme, işleme ve saklama sürecini içeren bir yapıdır. Bu sistemde, bilgiyi üretmek için üç temel aktivite bulunur: girdi, işlem ve çıktı. Girdi, organizasyonun içinden veya dış çevreden ham veriyi toplamayı ifade eder. İşlem, bu ham veriyi daha anlamlı bir formata dönüştürür. Çıktı ise işlenmiş bilgiyi insanlara veya kullanılacak diğer aktivitelere aktarır.

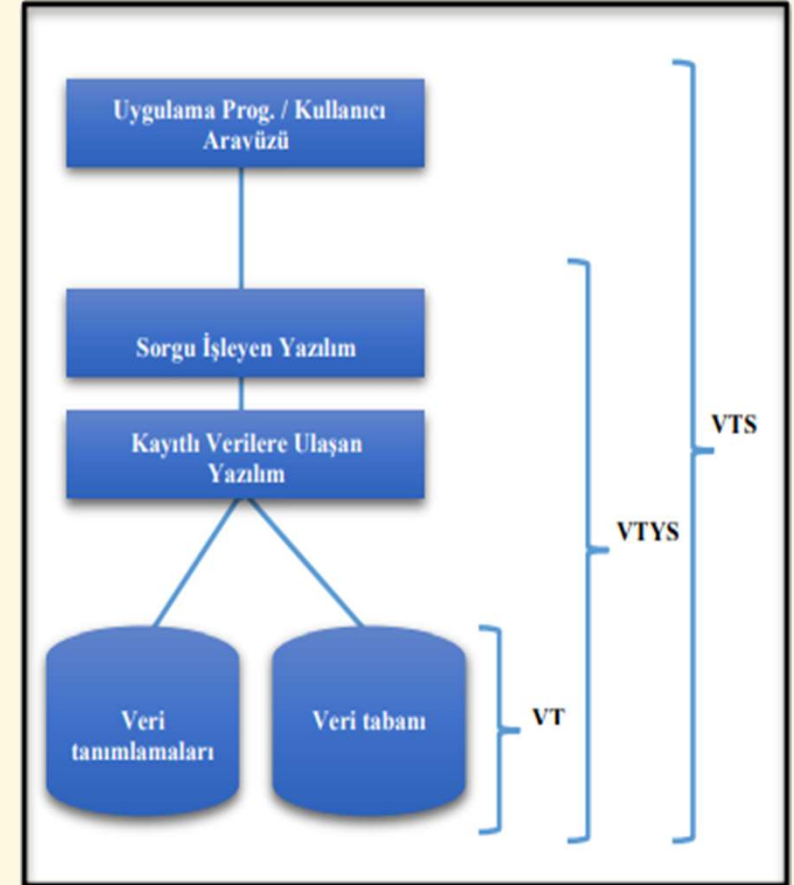
İşletmeler için, bilişim sistemleri sadece girdileri işleyip çıktı sağlayan mekanik yapılar değil, aynı zamanda daha derin anlamlar taşıyan yapılar olarak görülür. Bu sistemler, bilişim teknolojileri altyapısını kullanan yönetsel çözümlerdir. Bilişim sistemlerinden etkin bir şekilde faydalanabilmek için, organizasyonun hem yönetim, hem de teknoloji alanında hakim olması gerekir.

3-) Veri tabanı, belirli bir amaç doğrultusunda düzenlenmiş verilerin topluluğunu ifade eder. Bu veritabanlarında, birbirleriyle ilişkili veriler mantıksal ve fiziksel olarak tanımlanır ve depolanır. Gerçek dünyadaki nesnelerin ve ilişkilerin modellenmesini sağlarlar.

Veri tabanı yönetim sistemleri (VTYS), verilere aynı anda birden çok bağlantı sağlayabilme özelliği sunar. Bu sistemler, veri tabanının yönetimini sağlar ve verinin nasıl depolanacağını, nasıl kullanılacağını ve nasıl erişileceğini belirleyen bir kurallar sistemi oluşturur.

Veri tabanı, VTYS ve uygulama programlarını bir araya getiren yapıya "veri tabanı sistemi (VTS)" adı verilir. Bu sistem, kullanıcı ara yüzlerini içerir ve kullanıcıların verilere erişimini sağlar.

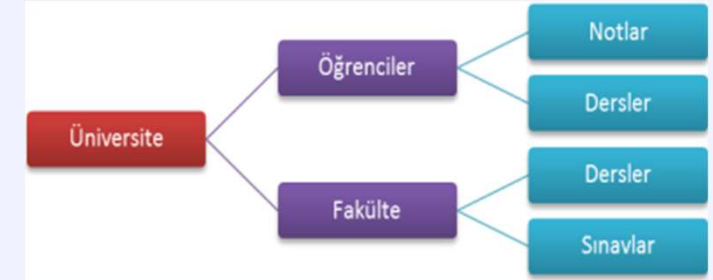
-Veri tabanı modellerini sekiz parçaya ayırabiliriz.



1-) Düz model veya tablo modeli, verilerin düz bir yapıda organize edildiği ve tablo biçiminde sunulduğu bir veri tabanı modelidir. Bu modelde, veriler iki boyutlu bir şekilde düzenlenir: sütunlarda benzer özelliklere sahip veriler bulunurken, satırlarda ise farklı veri grupları yer alır. Örneğin, kullanıcı adları ve şifrelerin saklandığı bir veri tabanı düz modelin bir örneğidir. Her satırda bir kullanıcıya ait şifre bilgileri bulunurken, sütunlarda aynı tipte veriler yer alır. Bu yapı, genellikle tek bir tablodan oluşur ve verilerin basit ve düzenli bir şekilde tutulmasını sağlar.

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

2-) Hiyerarşik Veri Modeli, ilk olarak 1960'larda geliştirilmiş ve adını verilerin depolanma yönteminden almıştır. Bu model, verilerin bir ağaç yapısı şeklinde düzenlendiği ve ilişkilendirildiği bir veri tabanı modelidir. Her bir veriye "kayıt" adı verilir ve bu kayıtlar bir ağaç yapısı içinde sıralanır. Kök olarak adlandırılan ilk kayıt, bir veya daha fazla alt kayıt içerebilir. Alt kayıtlar da kendi alt kayıtlarını içerebilirler. Her kayıt, bir ebeveyn kaydına sahiptir, ancak kök kayıt hariç. Bu model, verilerin hiyerarşik bir yapıda düzenlenmesini sağlar, böylece belirli bir sıra ve ilişki içinde tutulabilirler.



3-) Ağ veri modeli, 1970'lerin başlarında geliştirilmiş ve hiyerarşik veri modelinin geliştirilmiş bir versiyonudur. Bu modelin hızla benimsenmesinin nedeni, bir verinin doğal olarak diğer verilerle ilişkili olmasıdır. Ağ modelinin hiyerarşik modele kıyasla önemli bir farkı, bir düğümün içindeki verinin başka bir düğüme işaret edebilmesidir. Bu sayede, ağ modelinde birçok ilişki türü modellenebilirken, hiyerarşik modele göre daha esnek bir yapı sağlanır. Bu yapı aynı zamanda veri tekrarını azaltır, bu da veritabanlarının daha etkin ve optimize edilmiş olmasını sağlar.



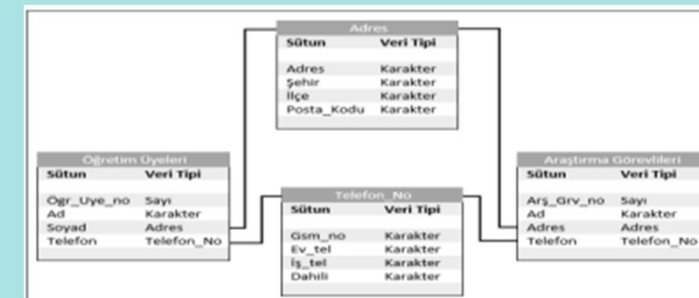
4-) İlişkisel Veri Modeli, çeşitlenen beklentilere cevap verme konusunda hiyerarşik ve ağ veri modellerinin yetersiz kaldığı zamanlarda ortaya çıkmıştır. Bu model, E. F. Codd'un 1970'te kaleme aldığı "A Relational Model of Data for Large Shared Data Banks" makalesi ile büyük bir ilerleme kaydetmiştir. İlişkisel veri modelinin merkezinde "ilişki" kavramı yer alır. Bu ilişkiler, veri içindeki bağlantıları modellemek için kullanılır. Dolayısıyla, ilişkisel bir veritabanı, çeşitli ilişkilerin bir araya gelmesinden oluşur. İlişkiler, genellikle satır ve sütunlardan oluşan iki boyutlu tablolarla ifade edilir. Her tablo için genellikle bir dosya bulunur ve her satır, birbirleriyle ilişkili verilerin bir gruplamasını temsil eder. Sütunlar ise bu verilere ait özellikleri içerir.



5-) Nesne Yönelimli Veri Modeli: Daha sonraları ortaya çıkmış ve başarısını kanıtlamıştır.



6-) Nesne ilişkisel veri modeli, nesne yönelimli özelliklerin ilişkisel veritabanlarının temel işlevselliği üzerine eklenmiş bir veritabanı modelidir. Bu modelde, veri tabanı nesneleri (nesnelerin nitelikleri ve davranışlarıyla birlikte) ilişkisel tablolarda depolanır ve ilişkiler aracılığıyla bağlantı kurulur. Oracle8, 1997 yılında piyasaya sürülen ve ilişkisel veritabanlarında nesne yönelimli özellikleri içeren ilk veritabanıdır.



7-) Çoklu ortam veri modeli, büyük medya dosyalarını işleme ve kullanıcıya adımları göstermeden işleme özelliklerine sahip olan bir veri tabanıdır. Dağıtık veri modeli ise verinin birden fazla bilgisayarda depolanıp ağ üzerinde dağıtıldığı bir veri tabanıdır.

4. VERİ TABANI TASARIMI

- Veri tabanı tasarımında gerçeklik, gereksinimlerin ve beklentilerin modellenerek veri tabanına aktarılması önemlidir. Kullanıcı gereksinimlerinin belirlenmesiyle başlanır ve veri grupları, tipleri ve depolama yapıları belirlenir. Gerçeğin veri tabanında anlaşılır bir şekilde temsil edilmesi için şemalar kullanılır. Kavramsal ve fiziksel düzeydeki şemalar farklı anlama mekanizmalarına yönelik olup, çeşitli veri modelleri kullanılır.
- Geleneksel veri tabanı tasarımı, kullanıcı düzeyinden başlayarak fiziksel düzeye adım adım ilerler. Kavramsal tasarımda, gereksinimlere göre kavramsal şema belirlenir. Kavramsal şema, veri tabanının genel yapısını tanımlar ve kullanıcıların veri tabanını anlamalarını sağlar. Fiziksel ayrıntılara girmeden varlıklar, ilişkiler ve kısıtlamalar üzerinde odaklanır. Kavramsal şema, donanım ve yazılımdan bağımsızdır ve kullanıcılar tarafından daha kolay anlaşılabilir.



- Kavramsal veri modelleri yüksek düzeyli oldukları için doğrudan gerçekleştirilemez. Bu nedenle, geleneksel veri tabanı tasarımında, kavramsal tasarımdan sonra bir veri tabanı yönetim sistemi seçimi yapılır. Veri tabanı yönetim sistemleri genellikle mantıksal veri modelleri kullanır. Bu nedenle, kavramsal şema, veri tabanı yönetim sisteminin veri modeline dönüştürülür. Bu dönüşüm işlemi bazen "mantıksal veri tabanı tasarımı" olarak adlandırılır.

- Fiziksel tasarım aşamasında, verinin en verimli şekilde nasıl fiziksel olarak düzenleneceği belirlenir. Bu aşamada iç şema oluşturulur, iç şema depolama yapılarını, kayıt formatlarını, alanları ve veri tabanına erişim yöntemlerini tanımlar. İç şema, fiziksel veri modellerini kullanarak yazılım ve donanıma bağımlıdır.

5. İLİŞKİSEL VE İLİŞKİSEL OLMAYAN (NoSQL) VERİTABANI SİSTEMLERİ

5.1-) İlişkisel veri tabanı sistemleri, tablolardan oluşur ve tablolar arasındaki ilişkilerle verileri yönetir. ACID ise bu sistemlerde sağlanan temel özellikleri ifade eder: bölünmezlik, tutarlılık, izolasyon ve dayanıklılık.

ACID; klasik ilişkisel veri tabanı sistemlerinde sağlanan temel özellikler:

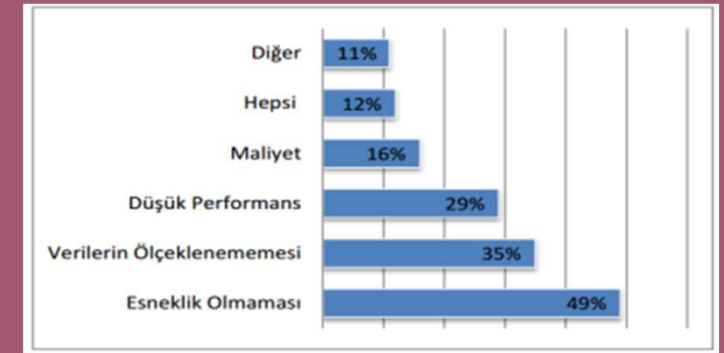
Bölünmezlik (Atomicity)

Tutarlılık (Consistency)

İzolasyon (Isolation)

Dayanıklılık (Durability)

- NoSQL, ilişkisel veri tabanı sistemlerine alternatif olarak gelişmiş ve genellikle büyük veri miktarlarını depolamak ve işlemek için kullanılan bir veri tabanı çözümüdür. Dünya çapında, büyük teknoloji şirketleri NoSQL'u tercih ederek büyük veri miktarlarını yönetiyorlar. Bu tür veri tabanları, ölçeklenebilirlik ve dağıtık veri depolama gibi özellikleriyle öne çıkar ve özellikle sosyal medya platformları gibi yüksek işlem hacmine sahip uygulamalar için uygun bir seçenek olarak görülür. Bu nedenle, NoSQL veri tabanları ilişkisel veri tabanlarının eksik kaldığı büyük veri işleme ve depolama ihtiyaçlarını karşılamak için tercih edilir.



- NoSQL veritabanları, Amazon'un "DynamoDB" ve Google'ın "BigTable" gibi isimlerle adlandırdığı sistemlerdir. İlişkisel veritabanlarının aksine, NoSQL tercihi hız, yatay büyüme ve gereksiz ek maliyetten kaçınma üzerine odaklanır. NoSQL veritabanları BASE (Basically Available, Soft state, Eventually consistent) prensiplerine dayanır ve esneklik sağlar.

- NoSQL veritabanları, e-ticaret, internet arama motorları ve sosyal ağlar gibi büyük ölçekli internet uygulamalarında güvenilirliğini kanıtlamıştır. Her bir NoSQL veritabanı çözümü, kendine özgü karakteristiklere sahiptir ve farklı iş gereksinimlerine uygun çözümler sunar.

Depolama	Protokol	Model	Lisans	Dil	
Memory mapped b-trees	BSON	Document	AGPL	C++	MongoDB
COW-BTree	HTTP/REST	Document	Apache	Erlang	CouchDB
Pluggable: InnoDB, LevelDB, Bitcask	HTTP/REST or TCP/Protocol buffers	Key/Value	Apache	Erlang	Riak
In memory, snapshot to disk		Key/Value	BSD	C++	Redis
Pluggable: BSV, MySQL, in-		Key/Value	Apache	Java	Voldemort
Memtable/SSTable	TCP/Thrift	Wide Column	Apache	Java	Cassandra
HDFS	HTTP/REST or	Wide Column	Apache	Java	HBase

Sadeleştirme	Arama	Esinlenilen
Evet	Evet	
Hayır	Hayır	Dynamo
Evet	Evet	Dynamo
Hayır	Hayır	
Hayır	Hayır	Dynamo
Evet	Evet	BigTable, Dynamo
Evet	Evet	BigTable

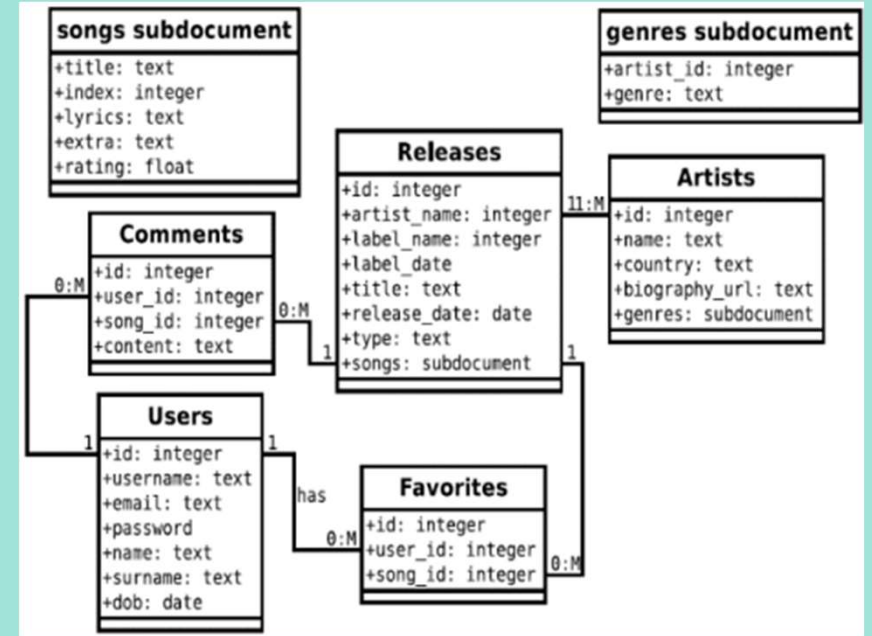
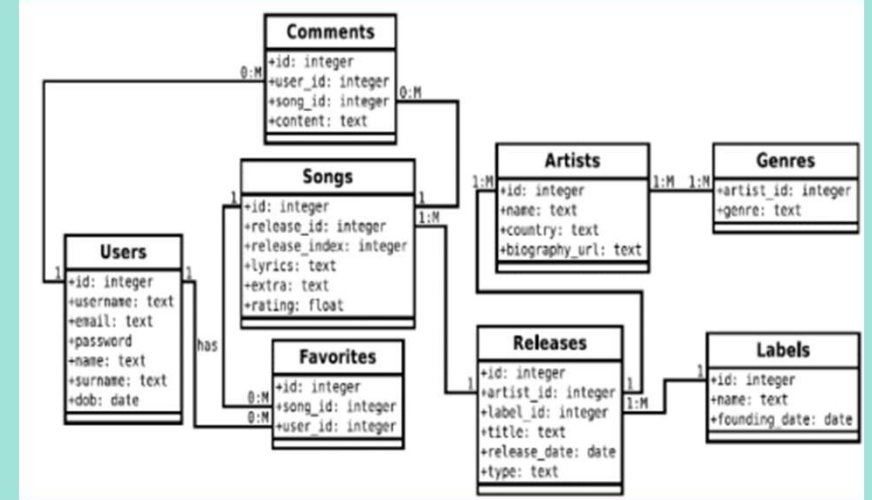
6. VERİTABANI MİMARİLERİNİN PERFORMANS KARŞILAŞTIRMASI

Bu çalışmada, MySQL ve MongoDB veritabanı sistemleri performans ve yatay ölçeklenebilirlik açısından karşılaştırılmıştır. MySQL, günümüzde en yaygın kullanılan ilişkisel veritabanı sistemlerinden biridir, MongoDB ise ilişkisel olmayan bir veritabanı çözümüdür ve yatay olarak ölçeklenebilme yeteneği sunar.

Çalışmada, her iki veritabanı sistemi için aşağıdaki işlemler gerçekleştirilmiştir:

1. Veri tabanı sunucu sistemleri özellikleri belirlenmiştir.
2. Veri tabanı şemaları oluşturulmuştur.
3. Sorgular belirlenmiştir.
4. Veri tabanı ayarları yapılandırılmıştır.
5. Ölçümler ve ölçüm metrikleri toplanmıştır.
6. Performans analizi yapılmış ve sonuçlar elde edilmiştir.

Bu çalışma, MySQL ve MongoDB veritabanı sistemleri arasındaki performans ve yatay ölçeklenebilirlik farklarını incelemeyi amaçlamaktadır.



Veri Tabanı Sorguları: Bu çalışmada üç farklı veri tabanı sorgusu kullanılmıştır. Birinci sorgu için sadece “SELECT” deyimi içeren basit bir sorgu hazırlanmıştır. İkinci sorgu için daha karmaşık “INNER JOIN” deyimi içeren bir sorgu hazırlanmıştır. Üçüncü sorgu için ise “SELECT” ile birlikte iç içe “JOIN”, “INNER JOIN” ve “WHERE” deyimi içeren detaylı karmaşık bir sorgu hazırlanmıştır.

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 ' AS rId  
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 '
```

Bu çalışmada ölçümler için zaman kavramı ön planda tutulmuştur. Üç farklı yöntem kullanılarak zaman ölçümleri gerçekleştirilmiştir:

- 1.Clock() fonksiyonu: Belirli bir süre boyunca CPU üzerinde harcanan zamanı ölçmek için kullanılmıştır.
- 2.Gettimeofday() fonksiyonu: Milisaniye hassasiyetiyle zamanlamaları sağlayarak sonuçların elde edilmesini sağlamıştır.
- 3.Slow Query Log (Yavaş sorgu kaydı): Her veritabanı, kendi yöntemini kullanarak zamanı ölçebilir. Bu yöntemde, önceden belirlenmiş uzun süren sorgular kaydedilir ve mikrosaniye hassasiyeti için yapılandırılabilir.

Ölçüm metrikleri, veritabanlarının performansını değerlendirmek için kullanılan ortak bir metrik sistemidir. En önemli faktör, bir görevi tamamlamak için gereken süre ve veritabanının bir işlemi tamamlama süresidir. Bu metrikler ayrı tutulmalı ve iyi anlaşılmalıdır. Çalışmada, aşağıdaki formül sorguların hesaplanması için kullanılmıştır.

$$\text{Saniyedeki Sorgu} = \frac{\text{Toplam Sorgu Sayısı} \times \text{Toplam İş Parçacığı Sayısı}}{\text{Ortalama Sorgu Süresi}}$$

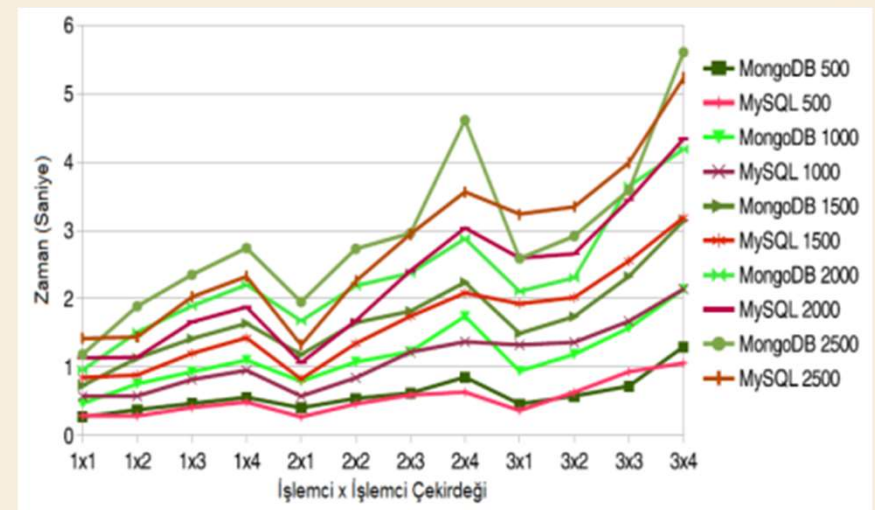
Her iş parçacığının saniye saniye sorgu başına nasıl tepki verdiğini ölçmek için aşağıdaki formül kullanılır.

$$\text{İş Parçacığı başına saniyedeki sorgular} = \frac{\text{Toplam Sorgu Sayısı}}{\text{Ortalama Sorgu Süresi}}$$

Bu çalışmada, veri tabanlarının farklı sorgu türlerine nasıl yanıt verdiği analiz edilmiştir. Hem okuma hem de yazma işlemleri için gerçekleştirilen farklı sorgu sayıları ve sonuçları şekillerle gösterilmiştir. Ayrıca, veri tabanı boyutunun performansa etkisi de incelenmiştir.

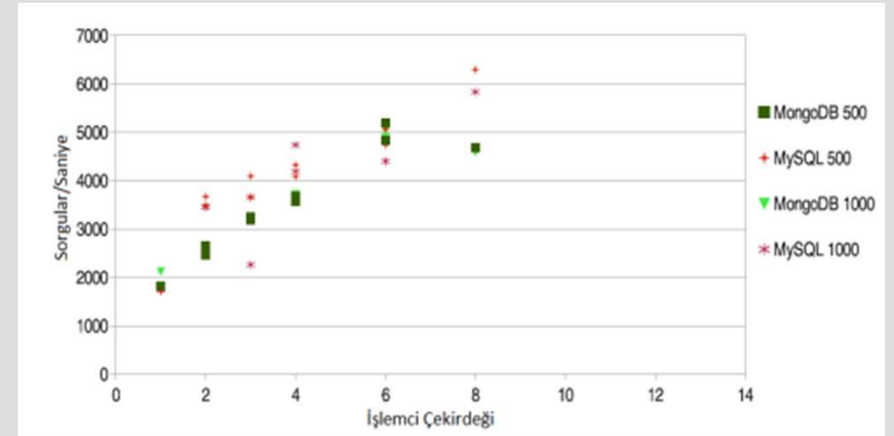
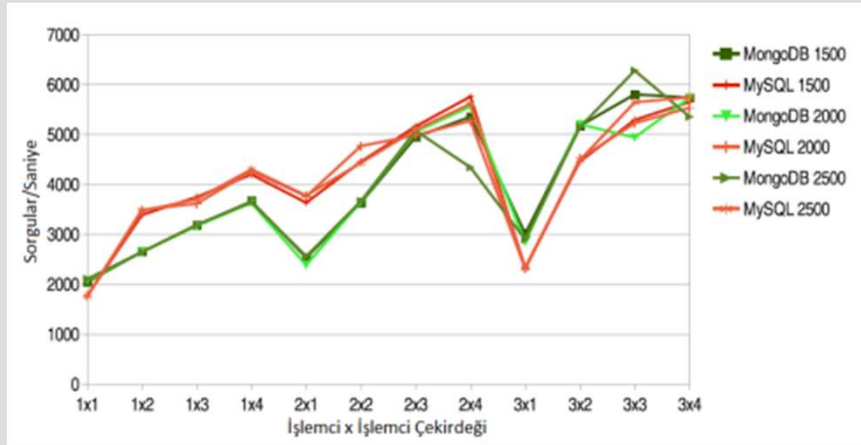
Çalışmada, veri tabanlarının detaylı olarak karşılaştırılabilmesi için çeşitli durumlar oluşturulmuştur. Ölçümler için farklı yapılandırmalar kullanılmıştır, bu yapılandırmalar işlemci sayısı ve çekirdek sayısı arasında değişmektedir. Ölçümlerde yapılan sorgu sayısı 500 ile 2500 arasında değişmektedir. Her bir ölçüm için beş adet test yapılmış ve her test sonucunda sorguların ortalama süreleri hesaplanarak raporlanmıştır.

MySQL ve MongoDB veritabanlarına yapılan basit bir sorgunun karşılaştırma testi uygulanmıştır. Yapılan analizde, sorgu sayısı arttıkça MongoDB'nin daha belirgin bir performans kötülüğü gösterdiği tespit edilmiştir. Bu karşılaştırma, işlemci çekirdeği sayıları toplamda aynı olduğunda (1x2 ve 2x1), 2 veya 1 işlemci kullanımının değişmediğini açıkça göstermiştir. MySQL veritabanı, özellikle 3 işlemci sayısı ile 1 işlemci çekirdeği sayısına göre incelendiğinde daha kötü performans göstermektedir.

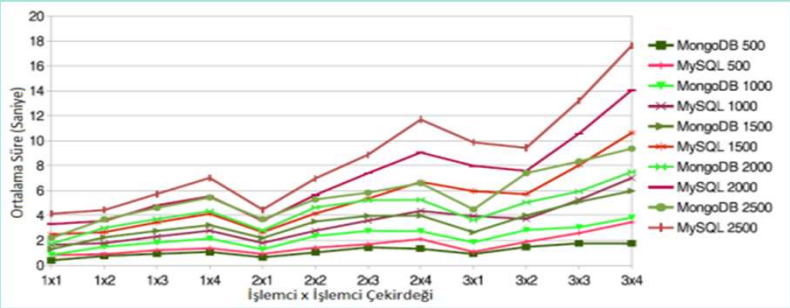


MySQL veri tabanı sisteminin, sorgu sayıları arttığında MongoDB üzerinde avantaj sahibi olduğu görülmektedir. Fakat 2 işlemci ve 3 işlemci çekirdeği yapılandırmasından sonraki diğer yüksek işlemci- işlemci çekirdeği sayılarında sorgu/saniye grafiğinde keskin bir şekilde azalma görülmektedir. MongoDB bu yapılandırmalarda daha fazla avantaj göstermiştir.

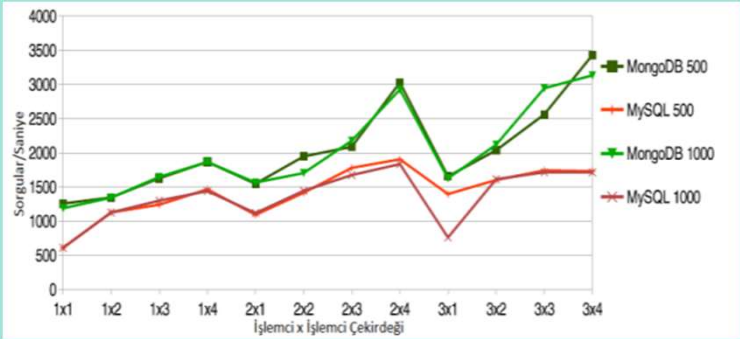
İşlemci çekirdeği miktarı ile saniye başına yapılan sorgu sayıları arasındaki ilişki analizi gösterilmektedir. MySQL için biraz daha iyi olan performans 4 işlemci çekirdeğine kadar hemen hemen aynıdır.



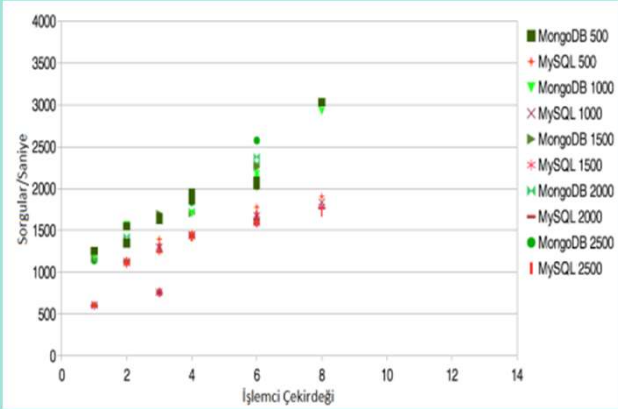
MySQL ve MongoDB veritabanlarına ikinci bir sorgu koduyla karşılaştırma testi uygulanmıştır. Yapılan analizde, sorgu sayısı arttıkça MySQL veritabanı sisteminin MongoDB'ye göre ortalama sorgu sürelerinde daha belirgin bir performans kötülüğü gösterdiği tespit edilmiştir.



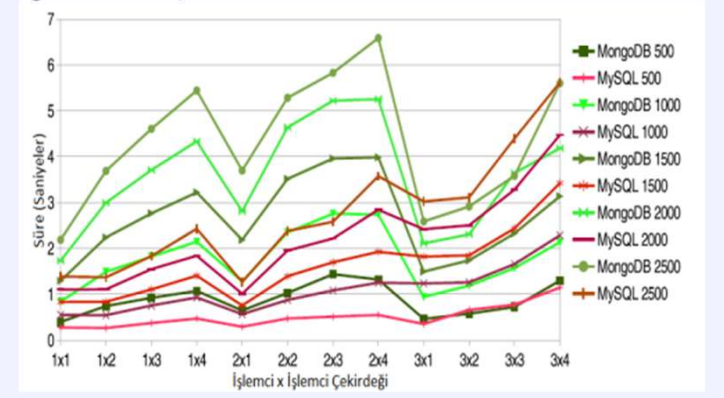
MySQL ve MongoDB veritabanlarına ikinci bir sorgu koduyla karşılaştırma testi uygulanmıştır. Bu test, küçük veri kayıtları (500 ve 1000 gibi) üzerinde gerçekleştirilmiştir. Yapılan analizde, MongoDB veritabanı sisteminin daha az sürede daha fazla sorgu yürütme yeteneğine sahip olduğu tespit edilmiştir. Sorgu sayısı değiştikçe performans ölçümü daha belirgin hale gelmiş ve MongoDB, sorgu/saniye başına %40 oranında daha iyi performans sergilemiştir.



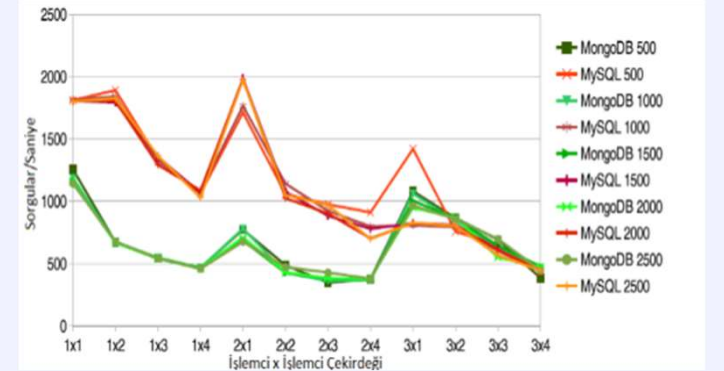
Şekilde, işlemci çekirdeği miktarı ile saniye başına yapılan sorgu sayıları arasındaki ilişki analizini göstermektedir. Bu analiz, ikinci sorgu koduyla yapılan karşılaştırma testine dayanmaktadır. Test, 500 ile 2500 veri kayıt setleri üzerinde gerçekleştirilmiştir. Yapılan analizde, MySQL veritabanı sisteminin veri kayıt miktarı ve sorgu sayısı arttıkça öncelikle kademeli bir düşüş ve ardından küçük performans artışları gösterdiği tespit edilmiştir. MySQL, artan işlemci çekirdeği sayısı ve veri kayıt miktarlarında neredeyse aynı sorgu/saniye performansı sergilemiştir. Öte yandan, MongoDB veritabanı sisteminin MySQL'e göre oldukça yüksek bir performans sergilediği gözlemlenmiştir. Aynı veri kayıt setlerinde MongoDB'nin MySQL'e kıyasla belirgin bir performans farkı ve avantajı olduğu görülmektedir.



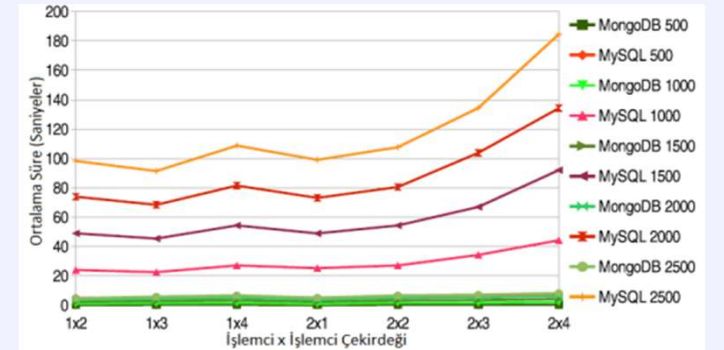
Şekilde, iç içe geçmiş "SELECT" ve "WHERE" işlemlerini içeren üçüncü bir sorgu sonucunda elde edilen performans değerleri gösterilmektedir. Yapılan analizlere göre, MySQL veritabanı sistemi MongoDB'ye göre daha iyi bir performans göstermiştir, özellikle veri kayıt sayısı farkı arttıkça. Ancak, işlemci ve işlemci çekirdeği yapılandırmalarının değiştirildiği durumlarda performans farklılıkları daha belirgin hale gelmiştir. Bu karşılaştırmada, işlemci ve işlemci çekirdeği sayıları 3x1, 3x2, 3x3 ve 3x4 şeklinde yapılandırıldığında, iki veritabanının neredeyse aynı performansı sergilediği gözlemlenmiştir.



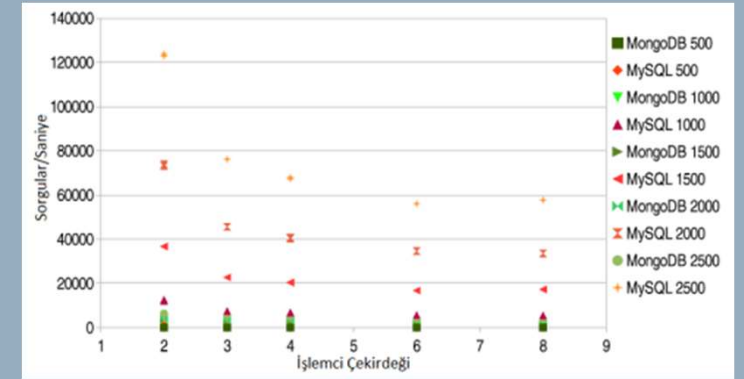
MySQL ve MongoDB veritabanlarına üçüncü bir sorgu kullanılarak uygulanan karşılaştırma testi sonuçları, Şekilde daha iyi anlaşılabilir bir şekilde gösterilmektedir. Analizlere göre, MySQL veritabanı sistemi 2x4 işlemci ve işlemci çekirdeği yapılandırmasında en iyi performansı sergilemektedir. Ancak, 2x1 ve 3x1 işlemci ve işlemci çekirdeği yapılandırmalarında her iki veritabanı için performans sorunları olduğu gözlemlenmektedir. Bu performans sorunu, özellikle MySQL'de daha belirgin bir şekilde ortaya çıkmaktadır.



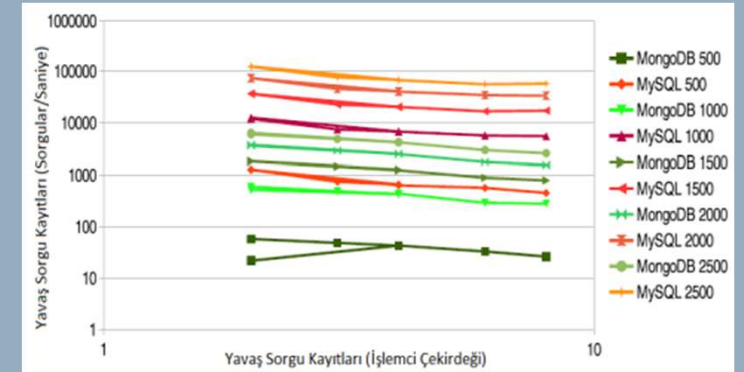
Şekilde üçüncü sorgu ile yapılan ortalama süre ölçümleri gösterilmektedir. Yapılan analize göre, MySQL veritabanı sistemi MongoDB'ye kıyasla belirgin bir performans kötülüğü göstermektedir, özellikle veri kayıt sayısı farkı arttıkça. Bu karşılaştırmada, eşdeğer veri kayıt setleri üzerinde gerçekleştirilmesine rağmen, MySQL veritabanının sorguları işleme ve sonuçlandırma süresi ortalamasının oldukça yüksek olduğu gözlemlenmektedir. Öte yandan, MongoDB daha belirgin ve istikrarlı bir performans sergilemektedir.



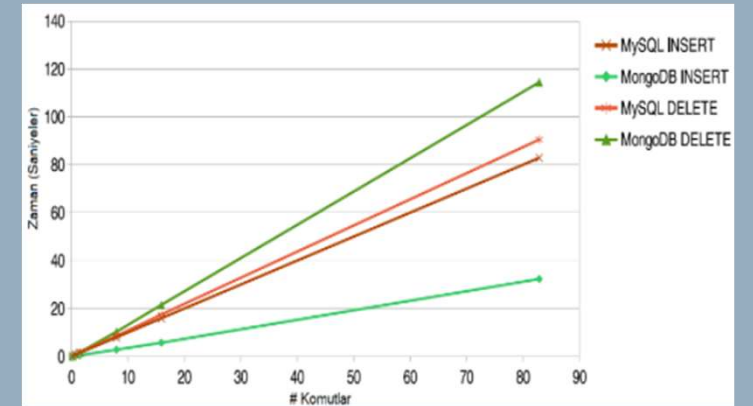
MySQL ve MongoDB veri tabanlarına üçüncü sorgu olarak tanımlanan detaylı ve karmaşık sorgu kodu içeren karşılaştırma testi analizi Şekilde gösterilmiştir. Bu test 500 ile 2500 veri kayıt setleri üzerinde yapılmıştır. MySQL veri tabanı sistemi, iki eksen boyunca logaritma kullanılarak çizilen grafikte logaritmik bir eğilim olduğu görüntüsü sergilemektedir. MongoDB'nin ise eğilimi nerede ve nasıl gösterdiği net olarak görülmemektedir.



Zamanlama ölçeği büyütülerek veri tabanları sistemleri arasındaki performans farkının Şekilde daha anlaşılabilir hale geldiği görülmektedir. Ölçek büyüdükçe MySQL'in performansındaki dezavantaj açıkça görülmektedir. MongoDB tüm veri kayıt setlerinde oldukça iyi bir performans gösterdiği ortaya konulmuştur.



MySQL ve MongoDB veritabanlarının performans analizinde, veri ekleme (INSERT) işlemlerinde MongoDB'nin MySQL'e göre daha iyi bir performans sergilediği görülmüştür. Veri silme (DELETE) işlemlerinde ise MongoDB ve MySQL benzer bir performansa sahiptir, ancak MySQL veritabanı sistemi daha fazla veri silme komutuyla daha iyi bir performans göstermektedir.



7.SONUÇ VE DEĞERLENDİRME

- Bu çalışmada, ilişkisel veri tabanları ile dağıtık mimariye sahip, SQL ve NoSQL olarak da adlandırılan veri tabanları karşılaştırılmış ve yönetim bilişim sistemleri açısından incelenmiştir. Amacı, yönetim bilişim sistemleri kapsamında veri tabanlarının modellenmesi ve niteliklerinin belirlenmesi, performans ölçümleri, süreç iyileştirmesi ve en uygun veri tabanı seçimi konularında kullanıcılara rehberlik etmektir. Son yıllarda teknolojik ilerlemelerle birlikte NoSQL kavramı ortaya çıkmıştır. Bu makalede, ilişkisel ve ilişkisel olmayan veri tabanı yönetim sistemlerinin performans karşılaştırması yapılmış ve farklı faktörlerin her bir veri tabanını nasıl etkilediği araştırılarak hangi teknolojinin hangi durumlarda daha uygun olduğu keşfedilmeye çalışılmıştır.
- Yapılan literatür taramasında, Ahmet Aladily'nin 2015 yılında yazdığı "En çok kullanılan NoSQL veri tabanları performans karşılaştırması" adlı tezde CouchDB, MongoDB, Cassandra ve HBase gibi dört farklı dağıtık veri tabanının performansı karşılaştırılmıştır. Yılmaz Gökşen ve Hakan Aşan tarafından hazırlanan "Veri büyüklüklerinin veri tabanı yönetim sistemlerinde meydana getirdiği değişim: NoSQL" adlı çalışmada ise NoSQL veri tabanları üzerine bilgiler verilerek esneklik, tutarlılık, performans, sorgulama dili ve veri bağlantıları gibi faktörler açısından ilişkisel veri tabanlarıyla genel bir karşılaştırma yapılmıştır. Bu çalışmada ise uygulanan analiz ile tutarlılık, esneklik gibi özelliklerin yanı sıra performans açısından farklı senaryolar oluşturularak detaylı bir analiz yapılmıştır.
- Çalışma hızı, yazılım rekabetinde kullanıcıların en önemli tercih kriterlerinden biridir. Bu nedenle, yaygın olarak kullanılan MongoDB ve MySQL veritabanı yönetim sistemleri, mümkün olduğunca eşit koşullarda işlem süreleri hesaplanarak performansları karşılaştırılmıştır. Farklı sorgu tipleri kullanılarak yapılan testlerde, detaylı ve karmaşık veri tabanı yapılandırmaları analiz edilmiştir.

- Yapılan analizler, NoSQL odaklı bir veritabanının büyük miktarda veri çiftlerini içerebildiğini ve basit bir şemaya sahip olduğu için veri çoğaltma konusunda MongoDB'nin daha hızlı ve karmaşık sorgu tiplerini çalıştırabildiğini göstermektedir. Farklı yapılandırmalarla yapılan performans testlerinde, ikinci sorgu tipiyle MongoDB'nin en iyi performansı sergilediği görülmüştür. Detaylı ve karmaşık sorgular üzerinde yapılan sonraki karşılaştırma testlerinde MongoDB'nin alt belge koleksiyonu kullanımı nedeniyle MySQL'e göre büyük bir avantaja sahip olduğu ortaya çıkmıştır. Bu avantaj, veri tekrarına izin verme maliyetiyle birlikte önemli bir şekilde görülmüştür. Bu tür sorgularda, büyük veri tabanı boyutundan kaynaklanan depolama ve bellek maliyetlerini hesaplarken NoSQL veritabanları gibi alternatiflerin göz önünde bulundurulması önemlidir.

- Yapılan son performans testleri, yazma ve silme işlemlerini içermektedir. Basit arama sorgularında veri silme işlemi dikkate alınarak yapılan karşılaştırmalar sonucunda MySQL veritabanı sistemi iyi bir performans sergilemiştir. Silme işlemi için verinin öncelikle bulunması gerektiği için silme işlemi doğrudan bağlantılanır. Her iki veritabanı da bu karşılaştırmada doğrusal bir eğilim gösterirken, MongoDB eklemeler sırasında MySQL'e göre belirgin bir şekilde daha iyi performans göstermiştir.

- Yapılan testlerin bir diğer önemli yönü, işlemci ve işlemci çekirdeklerinin farklı yapılandırmalarla nasıl kullanıldığının incelenmesidir. Bu çalışmada, veritabanlarının farklı sayıda işlemci ve çoklu işlemci çekirdeği üzerinde performanslarının test edilmesi ve karşılaştırılması için çoklu sorgulama işlemleri kullanılmıştır. Bu test, detaylı ve karmaşık sorguların kullanıldığı bir senaryo üzerinde gerçekleştirilmiştir. Sorgu karmaşıklığına bağlı olarak, veritabanları farklı davranışlar sergilemektedir. Bu farklılık, daha fazla sorgu sayısı ile ilişkili olarak sorguların/saniye analiz grafiklerinde gözlemlenmiştir.

- Yapılan analizler ve sonuçlar, işletmelere hangi durumlarda hangi veritabanı yönetim sistemini kullanmaları gerektiği konusunda fikir vermektedir. İki veritabanının da avantajları ve dezavantajları bulunmaktadır. İlişkisel veritabanı yönetim sistemlerinin kullanıldığı uygulamaların başlangıçta ilişkisel olmayan (NoSQL) sistemlere geçişinin zor olabileceği, veri kaybının söz konusu olabileceği ve NoSQL veritabanı sistemlerinin veri güvenliği alanında ilişkisel veritabanı yönetim sistemleri kadar gelişmediği gibi dezavantajları vardır. Ancak hız, geliştirme zamanı ve ölçeklenebilirlik gibi özelliklerle ilişkisel olmayan (NoSQL) veritabanılarının kullanılması, performans açısından daha etkin sonuçlar elde etmemizi sağlayacaktır.

SON

Levent ERGÖREN
02220224046