

**T.C.
SAKARYA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ**

**BULUT BİLİŞİM TEKNOLOJİLERİ VE NOSQL
VERİTABANLARI KULLANARAK TÜRKİYE’DE TERÖR
OLAYLARININ İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Kutlu EREN

Enstitü Anabilim Dalı: Yönetim Bilişim Sistemleri

Tez Danışmanı: Doç. Dr. Aykut Hamit TURAN

MAYIS – 2017

T.C.
SAKARYA ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ

**BULUT BİLİŞİM TEKNOLOJİLERİ VE NOSQL
VERİTABANLARI KULLANARAK TÜRKİYE’DE TERÖR
OLAYLARININ İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Kutlu EREN

Enstitü Anabilim Dalı: Yönetim Bilişim Sistemleri

“Bu tez/....../201.. tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.”

JÜRİ ÜYESİ	KANAATİ	İMZA
Doç. Dr. Aysel Hamit TURAN	BAŞARILI	
Doç. Dr. Abdülkadir HIZIROĞLU	BAŞARILI	
Yrd. Doç. Dr. Adem AKBIYIK	BAŞARILI	



SOSYAL BİLİMLER ENSTİTÜSÜ

YÜKSEK LİSANS/DOKTORA İNTİHAL YAZILIM RAPORU BEYAN BELGESİ

Tez Başlığı: BULUT BİLİŞİM TEKNOLOJİLERİ VE NOSQL VERİTABANLARI KULLANARAK TÜRKİYE’DE TERÖR OLAYLARININ İNCELENMESİ

Yukarıda başlığı belirtilen tez çalışmamın toplam 107 sayfalık kısmına ilişkin *Sakarya Üniversitesi Lisansüstü Yönetmeliği Madde 28* uyarınca aşağıda belirtilen filtrelemeler uygulanarak alınmış olan ve 25/04/2017 tarihinde Sosyal Bilimler Enstitüsü tarafından şahsıma iletilen *Turnitin* intihal tespit programı raporuna göre tezimin benzerlik oranı % 8’dir.

Uygulanan filtrelemeler:

- 1) Kaynakça hariç
- 2) Alıntılar dahil
- 3) 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Bu bilgiler doğrultusunda tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Öğrenci

(Adı – Soyadı, İmzası, Tarih)

ÖĞRENCİ BİLGİLERİ

Adı – Soyadı : Kutlu EREN
Öğrenci Numarası : 1260Y54114
Ana Bilim Dalı : Yönetim Bilişim Sistemleri
Programı : Yönetim Bilişim Sistemleri
Statüsü : ☐ Y. Lisans ☐ Doktora ☐ Bütünleşik Doktora

Danışman

(Adı - Soyadı, İmzası, Tarih)

ÖNSÖZ

Tez çalışmalarım süresince beni yönlendiren, teknik ve manevi anlamda benden desteklerini esirgemeyen danışmanım sayın hocam Doç. Dr. Aykut Hamit TURAN'a, yüksek lisans eğitimim sürecinde çok yardımı dokunan hocam sayın Yrd. Doç. Dr. Safiye SENCER'e, proje geliştirme aşamalarında beni destekleyen ve bana yardımcı olan değerli Kale Yazılım çalışanlarına, yapmış oldukları geri bildirimler ve yönlendirmeler için, tez savunma jürisinde yer alan hocalarım sayın Doç. Dr. Kadir HIZIROĞLU ve sayın Yrd. Doç. Dr. Adem AKBIYIK'a sonsuz teşekkürlerimi sunarım.

Hayatımın her aşamasında, eğitim süreçlerimde benden desteğini esirgemeyen, başta rahmetli babam olmak üzere sırasıyla anneme ve kardeşime sonsuz sevgi ve saygılarımı sunarım.

Kutlu EREN

12.05.2017

İÇİNDEKİLER

KISALTMALAR	vii
TABLO LİSTESİ	ix
ŞEKİL LİSTESİ	x
ÖZET	xii
SUMMARY	xiii
GİRİŞ	1
BÖLÜM 1: BULUT BİLİŞİM	7
1.1. Bulut Bilişim Nedir?	7
1.2. Bulut Bilişimin Tarihçesi	8
1.3. Bulut Bilişim Özellikleri ve Avantajları, Dezavantajları	10
1.3.1. Bulut Bilişim Avantajları	11
1.3.1.1. Sanallaştırma.....	11
1.3.1.2. Ölçekleme.....	11
1.3.1.3. Kullanıma Bağlı Ücretlendirme.....	11
1.3.1.4. Servis Düzeyi Anlaşması (Service Level Agreement – SLA).....	12
1.3.1.5. Dağıtılmış Veri Merkezleri.....	12
1.3.1.6. Bakım	12
1.3.1.7. Performans	13
1.3.8. Bulut Bilişim Dezavantajları	13
1.4. Bulut Bilişim Katmanları	14
1.4.1. İstemci	14
1.4.2. Uygulama.....	15
1.4.3. Platform.....	15
1.4.4. Altyapı.....	15
1.4.5. Sunucu	15
1.5. Bulut Bilişim Servisleri	15

1.5.1. Altyapı Hizmeti (IaaS – Infrastructure as a Service).....	16
1.5.2. Platform Hizmeti (PaaS – Platform as a Service)	16
1.5.3. Yazılım Hizmeti (SaaS- Software as a Service).....	17
1.6. Bulut Bilişim Yerleştirme Modelleri.....	17
1.6.1. Özel Bulut.....	17
1.6.2. Genel Bulut	17
1.6.3. Hibrit Bulut	17
1.7. Bulut Bilişim Teknolojileri	18
1.7.1. İnternet Teknolojileri	18
1.7.2. Dağıtık Hesaplama	19
1.7.3. Kaynak Sanallaştırma	19
1.7.4. Otonomik Hesaplama.....	19
1.8. Bulut Bilişim Platformları.....	20
1.8.1. Amazon Web Servisleri (AWS).....	20
1.8.1.1. Amazon S3 (Amazon Simple Storage Service).....	22
1.8.1.2. Amazon SQS (Amazon Simple Queue Service).....	22
1.8.1.3. Amazon RDS (Amazon Relational Database Service)	22
1.8.1.4. Amazon EC2 (Amazon Elastic Compute Cloud).....	23
1.8.1.5. Amazon ElasticSearch Service	23
1.8.2. Google Cloud Hizmetleri	23
1.8.3. Microsoft Azure Platforms	26
1.8.4. Diğer Bulut Hizmetleri	31
1.9. Bulut Bilişim Uygulamaları.....	32
1.9.1. SalesForce.com.....	32
1.9.2. Kissflow.com.....	33
1.9.3. Google Docs.....	34
1.10. Türkiye’de Bulut Bilişim	34
1.10.1. Paraşüt.com.....	37

1.10.2. Türk Telekom Bulut Ürünleri	38
1.10.3. Turkcell Akıllı Bulut.....	38
1.10.4. Netmera	38
1.10.5. PlusClouds.....	39
BÖLÜM 2: NOSQL (NON SQL, NOT ONLY SQL, NON RELATIONAL SQL) 41	
2.1. NoSQL Nedir?	41
2.2. NoSQL Özellikleri ve Avantajları, İlişkisel Veritabanları İle Karşılaştırması	42
2.3. NoSQL Veritabanı Çeşitleri.....	46
2.3.1. Anahtar-Değer Tipi Veritabanları	46
2.3.2. Sütun Bazlı Veritabanları.....	47
2.3.3. Döküman Bazlı Veritabanları.....	47
2.3.4. Graf Veritabanları.....	47
2.4. NoSQL Veritabanları Kullanım Alanları	47
2.5. Başlıca NoSQL Veritabanları	49
2.5.1. MongoDB.....	49
2.5.2. CouchDB	50
2.5.3. Cassandra.....	51
2.5.4. Amazon DynamoDB.....	52
2.5.5. SimpleDB	53
2.5.6. RavenDB.....	54
2.5.7. Google BigTable.....	54
2.6.7. Neo4J.....	54
2.6.8. Elasticsearch	55
BÖLÜM 3: LİTERATÜR TARAMASI.....57	
3.1. Bulut Bilişim.....	57
3.2. NoSQL	60

BÖLÜM 4: ÖRNEK UYGULAMA – TÜRKİYE’DE 1970-2014 YILLARI ARASINDA YAŞANMIŞ TERÖR OLAYLARININ MICROSOFT AZURE BULUT PLATFORMU ÜZERİNDE NEO4J NOSQL VERİTABANI MODELİ İLE AMPİRİK DEĞERLENDİRİLMESİ	63
4.1. Terör Verilerinin Elde Edilmesi	63
4.2. Elde Edilen Terör Verilerinin Analizi ve Sistem Modellemesi.....	64
4.2. Model Üzerinde Sorgulamalar.....	67
4.3. Microsoft Azure Bulut Platformunda NoSQL Çalıştırılması	70
4.4. Performans Analizi.....	72
SONUÇ VE DEĞERLENDİRME.....	744
KAYNAKÇA	833
EKLER.....	944
ÖZGEÇMİŞ.....	1013

KISALTMALAR

ADD-ON	: Eklenti
ACID	: Atomic, Consistent, Isolated, and Durable
API	: Application Programing Interface
AWS	: Amazon Web services
BASE	: Basically Available, Soft State, and Eventually Consistent
BLOB	: Binary Large Object
JSON	: Binary JSON
CRM	: Customer Relationship Management
CSV	: Comma Seperated Values
DBA	: Database Admin
DDOS	: Distributed Denial of Service Attack
GNU	: General Public License
GPU	: Graphics Processing Unit
HTML	: Hypertext Markup Language
HTTP	: Hypertext Transfer Protocol
HYM	: Hizmet Yönelimli Mimari
IAAS	: Infrastructure as a Service
IP	: Internet Protokol
JSON	: Javascript Object Notation
KVM	: Kernel-Based Virtual Machine
MMO	: Massive Multi-Player Online Games
PAAS	: Platform as a Service
REST	: Representational State

SAAS : Software as a Service
SBM : Server Message Back
SDK : Software Development Kit
SOA : Servis Oriented Artitecture
SOAP : Simple Object Access Protocol
SQL : Structured Query Language
VM : Virtual Machine
WSDL : Web Service Definiton Language
XML : Extensible Markup Language

TABLO LİSTESİ

Tablo 1 : Diğer Bulut Hizmetleri.....	31
Tablo 2 : Örnek NoSQL Tablosu.....	42
Tablo 3 : Örnek NoSQL Tablosu - 2	43
Tablo 4 : Örnek NoSQL Tablosu - 3	43
Tablo 5 : NoSQL Veritabanı Çesitleri.....	46
Tablo 6 : Veritabanı türleri.....	47
Tablo 7 : Veritabanı türleri ve dezavantajları	48
Tablo 8 : GDT CSV Verileri Önemli Sütun Başlıkları	64
Tablo 9 : Neo4j ve MySQL Performans Karşılaştırması	73
Tablo 10 : 1993-2011 Yılları Arasında Ölüm Verileri	79
Tablo 11 : 1984-2012 Arasında Jandarma Bölgesinde Yaşamını Yitiren Siviller.....	80

ŞEKİL LİSTESİ

Şekil 1 : Web Gelişim Semantiği	1
Şekil 2 : Veritabanı Gelişimi	4
Şekil 3 : Bulut Bilişim Genel Bakış	8
Şekil 4 : Bulut Bilişimin Zaman İçerisinde Gelişmesi.....	10
Şekil 5 : Bulut Bilişim Katmanları	14
Şekil 6 : Bulut Bilişim Servisleri.....	16
Şekil 7 : Bulut Bilişim Tipleri	18
Şekil 8 : Bulut Bilişim Platformları.....	20
Şekil 9 : Amazon Web Servisleri Yönetim Paneli	21
Şekil 10: Google App Engine Uygulama Sunucu Yapısı	24
Şekil 11: Google Compute Engine Ve Amazon Web Servis Karşılaştırması	25
Şekil 12: Microsoft Azure Platform Ana Panel.....	27
Şekil 13: Azure Makine Öğrenmesi İş Akışı.....	30
Şekil 14: Azure Makine Öğrenmesi İş Akış Tasarımı	30
Şekil 15: Salesforce Panel.....	33
Şekil 16: Hanehalkı Bilişim Teknolojileri Bilgisayar Kullanımı Temel Gösterler 2007-2016	35
Şekil 17: Paraşüt Panel.....	37
Şekil 18: Netmera Yapısı	39
Şekil 19: MongoDB ve MySQL Detaylı ve Karmaşık Sorgu Süre Analizi	45
Şekil 20: MongoDB ve MySQL Inner Join Karşılaştırması.....	45
Şekil 21: Mongo Yapısı	50
Şekil 22: Amazon Dynamo DB Yapısı	53
Şekil 23: Neo4J Genel Bakış	55
Şekil 24: GDT CSV Verileri Bazı Sütun Başlıkları.....	64
Şekil 25: Model Yapısı	66
Şekil 26: 2014 Yılında Ankara’da Gerçekleşmiş Ölümlü Terör Eylemleri.....	68
Şekil 27: 2005 ve 2014 Yılları Arasında Hakkari’de Gerçekleşen Bombalama/Batlama Tipindeki Saldırıları	69

Şekil 28: 1974 ve 2014 Yılları Arasında Bombalama/Patlama Tipinde Saldırıların En Çok Görüldüğü İlk 10 Şehir	70
Şekil 29: Azure Hizmet Modelleri ve Oluşturulan Sistem	71
Şekil 30: Azure Portal Yapılandırılmış Hali.....	72
Şekil 31: Neo4j ve MySQL Performans Karşılaştırması	73
Şekil 32: Bulut Bilişim ve Diğer Hizmetlerin Maliyet Karşılaştırması	76
Şekil 33: Dünyadaki Veri Artış Grafiği	78



Tezin Başlığı: Bulut Bilişim Teknolojileri ve NoSQL Veritabanları, Bulut Platformunda NoSQL Veritabanı Kullanarak Örnek Uygulama Geliştirilmesi	
Tezin Yazarı : Kutlu EREN	Danışman : Doç. Dr. Aykut Hamit TURAN
Kabul Tarihi : 12 Mayıs 2017	Sayfa Sayısı : xiii (ön kısım) + 94 (tez) + 7(ek)
Anabilimdalı : Yönetim Bilişim Sistemleri	Bilimdalı : Yönetim Bilişim Sistemleri Sistemleri
Bulut bilişim, şirketler için sunucu yükünü hafifleten ve maliyetini azaltan, erişilebilirlik süresi artıran, bilgiye istenilen zamanda ve çok kolay ulaşılabilmesini sağlayan, aynı zamanda değişen talebe bağlı olarak kolayca ölçeklendirilebilen, son zamanlarda birçok şirketin kullandığı sunucu teknolojileridir. Günümüzde gelişen internet sayesinde kullanıcıların kişisel bilgisayar, cep telefonu ve tablet gibi cihazlarda harcamış olduğu vakit çok artmıştır. Şirketler artan bu talebe cevap vermek zorunda olduklarından donanım ve yazılımlarını iyileştirmek zorunda kalmışlardır. Fakat değişen talepler yüzünden çok fazla sayıda sunucu ve yüksek trafik kullanmak optimum çözüm olmayabilir zira talepler değişken olabilmektedir. NoSQL ise internet kullanımındaki artış ile doğru orantılı olarak artan verilerin yönetilmesi için çözümler sunmaktadır. Klasik ilişkisel veritabanları çok büyük ve şema olarak değişken verilerin barındırılmasında sorunlar yaşamaktadır. Büyük verileri incelemek ve analiz etmek, analiz kod betikleri çalıştırmak ve bu verilerden anlamlı sonuçlar çıkarmak NoSQL veritabanları ile daha kolaydır. Bu çalışmada bulut bilişim teknolojiler ve NoSQL veritabanları incelenmiştir. Bu teknolojileri kullanarak hazırlanan uygulama, bu popüler teknolojilerin ne kadar efektif olduğunu araştırmaya yöneliktir.	
Anahtar Kelimeler: Bulut, Bulut bilişim, NoSQL, Veritabanı, Büyük Veri	

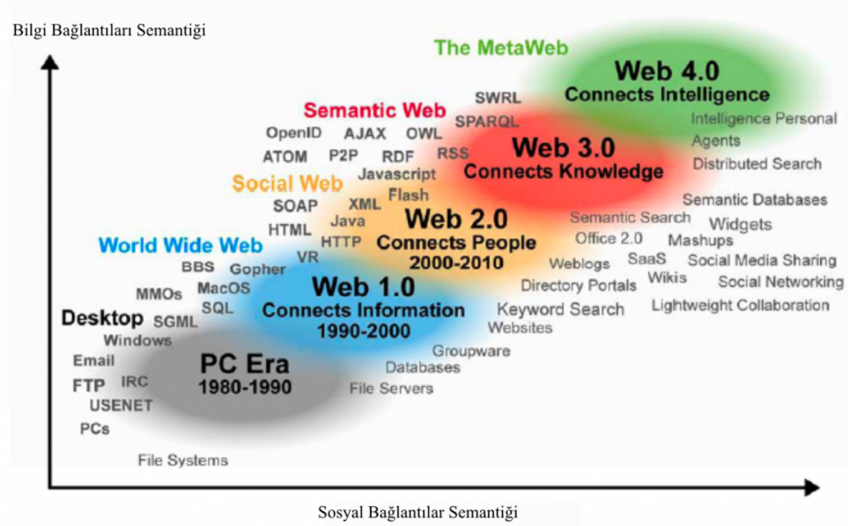
Title of the Thesis: Cloud Computing Technologies and NoSQL Databases, A Sample Application Using NoSQL Database On Cloud Platform	
Author : Kutlu EREN	Supervisor : Assoc. Prof. Aykut Hamit TURAN
Date : 12 May 2017	Nu. of pages : xiii (pre text) + 94 (main body)+7 (App.)
Department : Man. Info. Systems	Subfield : Management Information Systems
Cloud computing is a new technology to reduce hardware cost and traffic over servers. It increases accessibility time and enables people to reach the knowledge very easily. This technology is also very flexible to scale according to usage demand and recently become very popular and used by many companies across the world. Today, users are spending a lot of times on their mobile phones or tables. Companies had to improve their technologies by boosting their softwares and increasing the number of hardwares which they use in order to accommodate high network traffic and demand. However, increasing the number of hardwares is not always the best solution since the bandwidth demands and network traffic are always variable. Cloud computing is the solution for that kind of problems. Companies can instantly increase the number of servers or traffic according to their needs so they wouldn't have to take care of server maintenance. NoSQL offers a viable solution to handle the increasing data quantity as the people spend more and more time on the internet. Classical RDMS systems can face some problems handling large amount of schema variable data. Inspecting and analyzing the big data, running analysis scripts and interpreting results from analyzed data is much easier with NoSQL. In this work, cloud computing and NoSQL databases has been studied. The application has been implemented using these technologies aims to result how effective they are.	
Keywords: Cloud, Cloud Computing, NoSQL, Big Data, Database	

GİRİŞ

Bulut bilişim ve bulut hesaplamaları geçtiğimiz on yıldır bilişim sektörünün en önemli konseptlerinden birisi haline gelmiştir. Aslında geçmişi 1960'lara dayansa da günümüzde bilgi teknolojileri sektöründe önemli bir yer almıştır ve önemi her geçen yıl daha da artmaktadır.

1990'larda internetin ortaya çıkışı ile haberleşme alanında önemli gelişmeler de meydana gelmiştir. İnsanlar uzak noktalardan daha hızlı bir biçimde haberleşme olanağı bulmuştur. Bilgiye duyulan ihtiyacı karşılamak amacıyla yıllar içerisinde çeşitli teknolojiler geliştirilmiştir. Bu teknolojilere örnek olarak Web, SOAP, P2P teknolojileri gösterilebilir.

Bulut kavramını gerçekte bir telekomünikasyon terimi olup servis sağlayıcı ile son kullanıcı arasında kalan ağ üzerindeki sistemi sembolize eder. Bulut bilişim, builetişim kavramını, bünyesine sunucu bilgisayar sistemleri de ekleyerek genişletir (Sevli, 2011:9)



Şekil 1: Web Gelişim Semantiği

Kaynak: (Radar Networks Nova Spivack, 2007)

İnternetin bu denli gelişmesi ve kullanıcıların birbirlerine ve bilgiye daha kısa sürede ulaşma isteği yeni teknolojilerin doğmasına yol açmıştır. Hızlı veri alışverişine duyulan ihtiyaç zaman içerisinde “bulut bilişim” kavramının çıkmasını sağlamıştır.

Bulut bilişim temel olarak 1950’lere dayanmaktadır. Üniversiteler ve şirketlerde kullanılan büyük boyutlu bilgisayarlara istemci/terminal bilgisayar kullanarak ulaşılabilirdi. Bir ana bilgisayar alınması çok pahalı olduğundan mevcut ana bilgisayarın kaynaklarından en iyi şekilde faydalanılması çok önemliydi. Bu şekilde bilgisayar kaynaklarını en iyi şekilde kullanacak algoritmalar geliştirildi ve bulut bilişimin temelleri atılmıştır.

Anahtar bilgisayarlar, yarıiletkenlerin keşfinden önce var olan büyük ebatlı bilgisayardı. Her ne kadar günümüze eski bir teknoloji olsa da, anahtar hesaplama, bulut bilişim alanında pek çok fikrin gelişmesine katkı sağlamıştır. Bu büyük, yekpare sistemler, yüksek hesaplama hızı ve yedekli çalışma sistemlerine sahip olup, yüksek güvenilirlik ve erişilebilirlik sağlıyorlardı. Ayrıca, anahtar sistemler, son birkaç yılda yeniden canlanan sanallaştırma teknolojisinin de temelini oluşturmaktadır (Krishnan, 2010).

Günümüze bakıldığında donanım maliyetleri ucuzlamasına rağmen bilgiye ulaşım ihtiyacı da bir o kadar artmaktadır. Şirketler kullanıcılara en iyi hizmeti vermek amacıyla kendi bulut yapılarını oluşturmakta ya da dünyada bulut hizmeti sağlayan şirketlerden altyapı satın almaktadır. Özellikle platform bağımsız bilgi paylaşımı bulut bilişimin kullanımının artmasında önemli rol oynamaktadır.

Günümüz bilgi işlem yapıları buluta doğru genişlemektedir. Servis mimarisine dayalı bulut bilişim; veri ve uygulamalara, cihazdan bağımsız olarak erişilebilmesini, uygulamaların yaygınlaştırılıp kolaylıkla ölçeklendirilebilmesini sağlamaktadır (Sevli, 2011:3)

Veri sunulmasındaki sağladığı hız ve kaynak kullanımındaki etkili yapısı ile bulut bilişim artarak kullanılmaya devam edecektir. Şirketler sistemlerini bulut üzerine taşıyarak yönetimlerine hızlı ve platform bağımsız uygulamalar sunabilmektedir. Bu sayede yöneticilerin karar alma süreleri artmaktadır. Bir yönetici cep telefonundan

inceleme yaparak onay verebilmekte, raporları inceleyebilmektedir. Son kullanıcı diye tabir edilen uygulama kullanıcıları verilerine heryerden erişebilmekte ve bu verileri yönetip, paylaşabilmektedir.

Bulut'un sağlamış olduğu bu kolaylıklar önümüzdeki yıllarda birçok şirketin buluta geçmesine, altyapılarının ve verilerinin büyük bölümünü buluta taşınmasını sağlayacaktır.

Veritabanı sistemleri geçmişten günümüze veri depolama işlemleri için önemli yer tutan araçlardır. Temelleri 1960'lı yıllara dayanmaktadır. Bu yıllarda veriler direk erişimli depolama alanlarına kaydedilmekteydi. Bu dönemde veri modeli ağ modeli ve hiyerarşi modeli olarak 2 şekilde tanımlanmıştı. 1970'lere gelindiğindeyse bir IBM çalışanı olan Codd (1970), yayınlamış olduğu araştırmada ilişkili veri tabanlarına ilk olarak değindi. Bu yayınında ilişkili listeler yerine sabit uzunluktaki tablo yapısından bahsetti.

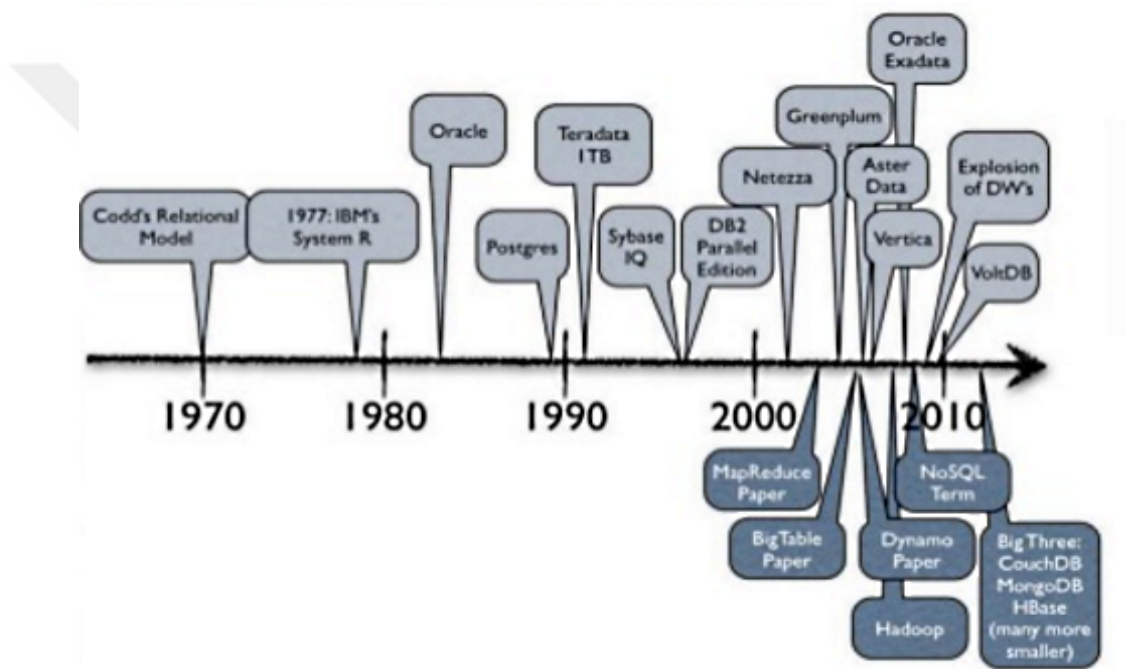
1970'lerin sonunda gelindiğinde IBM tarafından daha sonra Database2 (DB2) olarak anılan veritabanı geliştirildi. Oracle, IBM'in yayınlamış olduğu "*System R*" dökümanını baz alarak 1978'de yayınlandı. 1980'lerde ise masaüstü depolama sistemleri geliştirildi. Aynı zamanda bu yıllarda SQL standart sorgulama dili halini aldı. 1990'lara gelindiğinde nesne tabanlı programlama ile birlikte yazılımcılar verilerini kullandıkları programlar içerisinde nesne olarak yer vermeye başladılar.

2000'li yıllara gelindiğinden ilişkisel veritabanlarının sahip oldukları eksiklikleri gidermek amacıyla NoSQL veritabanı kavramı türetili. IBM (2015:1) 2015 yılındaki yayınında, yazılımcıların ve bilişimcilerin, verileri ilişkisel veri modeline entegre ederken aynı zamanda performans için dinamik bir ölçekleme yapmada çok zorlandıklarını, hatta bazen bu işlemin imkansız olduğunu belirtmiştir. Bu tespit ilişkisel veritabanlarının eksiklerini özetlemektedir.

NoSQL veritabanı 2009 yılından itibaren "NoSQL" olarak adlandırılmaya başlandı. 2000'li yıllardan sonra çeşitli NoSQL veritabanları ortaya çıktı. 2006 yılında Google, yayınladığı dökümanla Bigtable dağıtık yapılı veritabanını duyurdu. Google Bigtable'ı binlerce sunucu üzerinde çalışabilen ve petabyte'larca veriye ölçeklenebilen dağıtık bir

depolama sistemi olarak tanımlamıştır (Fowler, 2015:8). Amazon 2007 yılında kendi dökümanını yayınlayarak Dynamo veritabanını duyurmuştur. Bununla birlikte çeşitli hizmetlere yönelik olarak farklı yapılarda veri saklama özelliğine sahip birçok veritabanı ortaya çıkmıştır. NoSQL veritabanları büyük verileri işleme de çok iyi performans sağlamaktadırlar. Aynı zamanda dağıtık çalışabilmektedirler ve bu özelliği NoSQL veritabanlarını çok kullanışlı hale getirmektedir.

Şekil 2’de veritabanları gelişimi gösterilmiştir.



Şekil 2: Veritabanı Gelişimi

Kaynak: (Faiveley Transport-SlideShare, 2015)

NoSQL veritabanları günümüzde halen gelişmekte olan teknolojilerdir. Faydaları zamanla kullanıldıkça anlaşılacaktır. NoSQL veritabanları her soruna çözüm olarak geliştirilmemiştir. Veritabanı seçimi yaparken ihtiyaçların net olarak belirlenmesi büyük önem arz etmektedir. NoSQL veritabanı çeşitleri iyi analiz edilirse bir çok sorun için iyi çözümler getirdikleri anlaşılacaktır.

Çalışmanın Konusu

Çalışmanın konusu, günümüzde kullanılan en popüler bulut bilişim platformları, NoSQL veritabanları araştırılmış ve literatürdeki çalışmalar incelenmiştir. Yapılan araştırmalardan sonra bulut platform üzerinde NoSQL veritabanı kullanılarak örnek bir uygulama oluşturulacaktır. Bu uygulama ile bulut bilişim ve NoSQL veritabanlarının getirdiği kolaylıklardan faydanılması hedeflenmiştir.

Çalışmanın Önemi

Bulut bilişim ve NoSQL veritabanları gün geçtikçe önemi artan kavramlardır. Özellikle büyük verilerin depolanması ve bu verilerin analiz edilmesinde oldukça başarılıdırlar. Bunun yanı sıra bu işlemleri daha düşük maliyetlerde yapma imkanı sunmaktadırlar. Literatür taraması yapıldığında bulut bilişim alanında çeşitli çalışmalar yapılmıştır ancak NoSQL veritabanları ile Türkiye’de çok fazla uygulama yapılmadığı görülmüştür. Bu yaklaşımdan hareketle bulut bilişim ve NoSQL teknolojilerinin genel özellikleri ve ne işe yaradıkları, bu teknolojilerin teknik yapıları, kullanım alanları, avantajları dezavantajları, performansları ve çeşitleri detaylı bir biçimde incelenmiş, bu teknolojiler kullanılarak ülkemizin en büyük sorunlarından birisi olan terör sorununa yaklaşım amacıyla örnek uygulama geliştirilmiştir.

Çalışmanın Amacı

Bu çalışma kapsamında ulaşılmak istenen amaç; *“1970-2014 yılları arasında Türkiye’de gerçekleşmiş terör olaylarının Graph veri tabanı ile modellemesi ve bulut platformunda çalıştırılmasıdır”*. Bu amaca ulaşmak için dünyadaki bulut bilişim ve NoSQL veritabanı teknolojileri detaylıca incelenmiştir. Ayrıca *“ülkemizde bulut bilişim ve NoSQL veritabanlarının kullanımının yaygınlaştırılması”, “kullanıcıların donanım maliyeti azaltılarak harcanan enerjinin azaltılması ve doğa dostu bir altyapı kurmaları”, “çalışmanın genel olarak bilgi içeren bir döküman olarak bulut bilişim ve NoSQL veritabanları hakkında bilgi sahip olmak ve bu teknolojilerden faydalanmak isteyen kişilere kılavuz olması”, “yapılan uygulamada kullanılan verilerin artırılması,*

modelin genişletilmesi ve akademik olarak bir karar destek sisteminin geliştirilmesi” hedefler arasındadır.

Çalışmanın Yöntemi

Terör olaylarının sonuçları göz önünde bulundurularak bulut bilişim ve NoSQL teknolojileri araştırılacak ve bu teknolojilerden faydalanarak terör olaylarını inceleyen örnek uygulama geliştirilmiştir. Bu uygulama ile 1970-2014 yılları arasında Türkiye’de gerçekleşmiş terör olaylarının Graph veri tabanı ile modellenmesi yapılmıştır. Bu modelleme bulut platformunda çalıştırılmıştır. Bu model yardımı ile verilerden görsel modeller oluşturularak olayların yorumlanması ve ileride gerçekleşebilecek olaylara yönelik kolluk güçlerinin erken önlem alınmasına, model üzerinde aramalar yapılarak bilgiye daha doğru ve hızlı ulaşılması katkı sağlanmıştır. Güncel teknolojiler olan bulut bilişim ve NoSQL veritabanlarının gücünden faydalanılarak efektif bir uygulama geliştirilmiştir. Yapılan tez çalışması bölümleri aşağıdaki gibidir;

- **Bölüm 1:** Bulut bilişim, bulut bilişim kavramları ve yapısı, avantajları-dezavantajları, çeşitli bulut bilişim platformları, Türkiye’de bulut bilişim kısımları yer almaktadır.
- **Bölüm 2:** Nosql, nosql kavramları ve yapısı, avantajları ve dezavantajları, nosql veritabanı çeşitleri, çeşitli nosql veritabanı platformları incelenmiştir.
- **Bölüm 3:** Bulut bilişim ve nosql ile ilgili yapılmış çalışmaların incelenmesi yapılmıştır.
- **Bölüm 4:** Bulut platform üzerinde nosql veritabanı kullanılarak Türkiye’de 1974 ve 2014 yılları arasında gerçekleştirilen terör eylemlerinin modellenmesi ve incelemesine yer verilmiştir.
- **Sonuç ve Değerlendirme:** Bulut bilişim ve nosql veritabanları değerlendirilerek yararlarına vurgu yapılmış, örnek uygulama ile gerçekleşebilecek terör olaylarına karşı önlem alınmasını sağlayacak bir sistem ortaya konulmuş, gelecek çalışmalar için akıllı bir sistemin yapılması amacı ortaya konuşmuştur.

BÖLÜM 1: BULUT BİLİŞİM

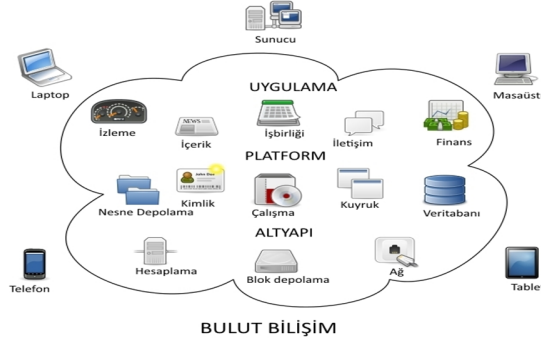
1.1. Bulut Bilişim Nedir?

Bulut bilişim hali hazırda gelişen ve ilerleyen bir kavram olduğundan bulutu genel bir tanım altında değerlendirmek zordur. Bilişim sektöründeki unsurlar bulutu kendi bakış açılarından yorumlamaktadır. Yinede yapılan tanımlamalar birbirine paraleldir.

Bulut, çok sayıda bilgisayarın birleşimi ile meydana gelen veri merkezlerini içeren, kullanıcıların kaynaklara internet üzerinden erişimine imkân veren bir bilgi işlem ağıdır (Sevli, 2011:8). Bulut hizmeti sunan şirketler sistemlerine erişimlere, kendi geliştirmiş oldukları web platformları üzerinden izin verir. Bulut bilişim kavramı bu noktada devreye girmektedir. Şirketler kullanıcılara sunmuş oldukları sistemler kaynakların dağıtılması, bilgi üzerinde işlem yapmayı sağlar ki buna bulut bilişim denir.

Bulut bilişim, kendi sisteminde kullandığı bütün donanım ve yazılım unsurlarında sistemi kullananlar için soyutlama kavramını sunmaktadır. Soyutlaa kavramı sayesinde kullanıcılar kendi alt yapılarında bulunan donanım ve yazılımları hizmet olarak almalarından dolayı, detaylı olarak konfigürasyonu bilmesi gerekmemektedir (Çatak, 2014:5).

Bulut bilişim, bilgisayarların yeteneklerini genişleten, kullanıcıların bir dizi yazılım ve servise internet üzerinden erişimlerine olanak tanıyan bir teknolojidir (Rayport ve Heyward, 2009:4).



Şekil 3: Bulut Bilişim Genel Bakış

Kaynak: (SuitSupply, 2016:1)

Bulut bilişim, depolama ve veri işleme yükünün istemci bilgisayarlar üzerinden alıp, uzak sunucular üzerine aktaran bir modeldir. Sunucular üzerinde barındırılan ve işleme tabi tutulan veriler, ihtiyaç duyulduğu anda, internet üzerinden istemci bilgisayara aktarılır (Sevli, 2011:8). Bu anlamda istemci kullanıcı açısından herhangi bir fark arzetmemektedir. Kullanıcı sanki herhangi bir sunucuya bağlıyormuş gibi düşünülebilir. Bağlandığı sunucu tek bir bilgisayar değil, bir bilgisayarlar bütünüdür.

Dünyada bulut bilişim alanında birçok firma yer almaktadır. Bu firmalar gerek alt yapı gerekse yazılım kategorilerinde hizmetler vermektedirler. Bulut denildiğinde dünyada akla gelececek Amazon şirketinin sunmuş olduğu Amazon Web Servisleri'dir. Amazon 2008 yılından sektöre giriş yapmıştır. Google ise, Google Cloud Platform adında bulut hizmeti sunmaktadır. Microsoft, Azure bulut platformu ile 2010 yılında sahaya giriş yapmıştır. Teknik donanın ve birçok yazılım hizmeti sunmaktadır. Bu şirketlerin yanı sıra IBM, Apple, Rackspace, Salesforce gibi birçok şirket bulut bilişim ürün ve hizmetleri sunmaktadır.

1.2. Bulut Bilişimin Tarihçesi

Bulut bilişim tarih olarak 1950-1960'lara dayanmaktadır. Yapay zeka çalışmalarında öncü ve LISP programlama dilinin yaratıcısı John McCarthy'nin 1960'larda söylediği "Bir gün hesaplama işlemleri kamusal ağlar üzerinde çalışacak" sözü bulut bilişimin geçmiş dönemlere dayandığını gösterir.

Bulut biliřim, bir ana bilgisayar alınmasının pahalı olduđu d nemlerde mevcut bilgisayarın en verimli řekilde zaman paylařımlı olarak kullanıcıların bilgisayar kaynaklarını en iyi bi imde kullanmasını  n g ren fikire dayanmaktadır.

Google y netim kurulu bařkanı Eric Schmidt 1993 yılında “Ađlar, bilgisayarlar  zerindeki iřlemciler kadar hızlı  alıřır hale geldiđinde, bilgisayarlar t m iřlemlerini ađ  zerinde yapmaya bařlayacaklar, ađ  zerinde yayılacaklar” s z yle bulut biliřimin gelecekteki yerini belirtmiřtir.

Bulut biliřimin geliřmesinde bir ok etken rol oynamıřtır. Bu geliřime en b y k etken olarak geliřen bilgisayar teknolojileri g sterilebilir. Geliřen elektronik sayesinde kullanıcıların bilgisayar kullanımı artmıř ve bulut biliřimin geliřiminde mesafeler alınmıřtır.

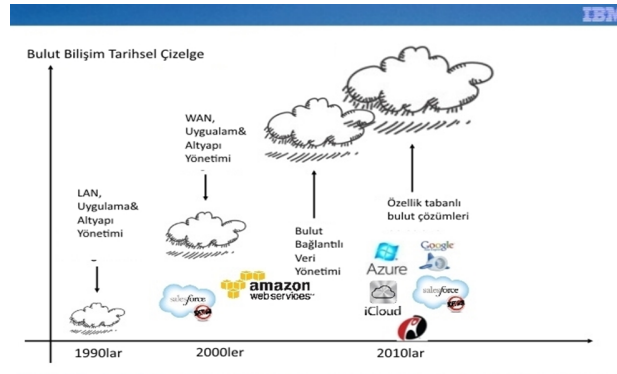
Yarı iletkenlerin icadı ile bilgisayar boyutlarında k   lmeler meydana gelmiř ve performansları artmıřtır. B ylece ana atı bilgisayarın kullanımı azalmıřtır. Ana atı hesaplama ve bulut biliřim, kaynakların merkezileřtirilmesi ve bu sayede, alımı ve bakımı son derece pahalı olan sistemlerin kiralanarak kullanılması fikri a ısından birbirlerine benzemektedirler (Sevli, 2011:9)

Zaman paylařımı, bilgisayar iřlemcisinin giriř/ ıkıř iřlemlerini bekleyerek harcadıđı zamanı diđer kullanıcılara tahsis etmesini ve bu sayede zamanın paylařımlı ve verimli kullanımını  ng r r (Krishnan, 2010:2). Bu sayede bilgisayar kaynakları farklı kullanıcılar arasında paylařtırılabilmektedir ve bu sayede bulut biliřim geliřmesine katkıda bulunmuřtur.

Virtual Machines (VM) (Sanal Makina) kavramı 1970’lerde geliřtirilmiřtir. Bu sayede bir bilgisayar  zerinde parallel olarak birden fazla iřletim sistemi  alıřtırılabilmesi fikri  ng r lm řt r. Dađıtık hesaplama sistemleri 1990’larda ortaya  ıkmıřtır ve bilgisayarların bir elektrik ađına benzer bir řekilde birbirine bađlı olarak  alıřmalarını sađlayan teknolojidir.

Günümüzde 2 milyardan fazla bilgisayar, tablet ve cep telefonu kullanılmakta ve bu sayı her geçen gün artmaktadır. İnternet kullanımının yaygınlaşması ile birlikte amazon.com bulut bilişim yapısının gelişmesinde önemli bir rol oynamıştır. 2006 yılında Amazon, kurduğu bulut sistemine Amazon Web Servisleri (AWS) üzerinden erişim olanağı sağlamıştır. Bulut bilişim konusunda sunduğu sistemler ile bu sektörde öncü şirketlerden biridir. 2007 yılında Google, IBM ve birçok üniversite bulut bilişim projeleri üzerinde çalışmalara başlamıştır.

Merkezleştirilmiş bir kaynak olarak bulutun kapasitesi, endüstriyel ölçeklerde olacak biçimde inşa edilebilir. Bunun anlamı, yüzlerce hatta binlerde yüksek performanslı bilgisayarın birleştirilerek, büyük bir işlem gücü oluşturmasıdır. Bu teknolojinin internet üzerinden istek ve ihtiyaca bağlı olarak kullanılabilir hale getirilmesi ile bulut, dünya üzerindeki bireysel kullanıcılara, şirketlere, organizasyonlara olağanüstü bir isle gücüne istedikleri herhangi bir cihaz ve konumdan rahatlıkla erişebilme vaadini yerine getirmiş olmaktadır (Sevli, 2011:14). IBM'in 2011 de yayınladığı Şekil 4 bulut gelişimini özetlemektedir.



Şekil 4: Bulut Bilişimin Zaman İçerisinde Gelişmesi

Kaynak: (IBM, 2011:8)

1.3. Bulut Bilişim Özellikleri ve Avantajları, Dezavantajları

Bulut bilişim birçok özelliği barındırmaktadır. Bunlardan bazıları şöyledir :

1.3.1. Bulut Bilişim Avantajları

Bulut bilişim temele az donanım ile daha çok hizmet alabilme felsefesi üzerine kurulmuştur. Hizmet alırken ise tutar maliyetinin en aza indirilmesi amaçlanmıştır. Bu açıdan bulut bilişimin kullanıcılarına getirdiği birçok avantaj bulunmaktadır. Bunlar aşağıda sıralanmıştır:

1.3.1.1. Sanallaştırma

Sanallaştırma, fiziksel bir kaynağı istenilen sayıda mantıksal parçaya bölerek, kaynağın verimliliğini artırmaya yönelik bir konsepttir. Sanallaştırma yazılımları, sanal makinelerle ortak olarak donanımları paylaşır. Bu paylaşım ne kadar iyi yapılırsa, o kadar fazla verim alınacaktır. Maliyet o oranda azalacak ve sistem performansı da o oranda artacaktır (Demircioğlu, 2016:2).

Sanallaştırma, bulut bilişimde çok önemli bir yere sahiptir. Bu sayede yüzlerce fiziksel makina, binlerce kullanıcıya hizmet verebilecek hale gelmektedir. Bu bulut hizmet sağlayıcı firmalar için çok önemli bir avantajdır.

1.3.1.2. Ölçekleme

Bulut bilişim platformları kullanıcıların talebine bağlı olarak ölçeklendirilebilir. Hizmet sağlayıcıları sanallaştırma sayesinde müşterilerine anlık olarak kaynak sağlayabilmektedir. Otomatik ölçeklendirme, kullanıcının belirlediği kriterlere göre aşağı ya da yukarı yönlü sistemin otomatik olarak ölçeklenmesi özelliğidir (Kriushanth, Arockiam ve Mirobi, 2013:3). Yani bir firma bulut sağlayıcısı arayüzlerinden kullanıcı bant genişliğini, depolama alanını, işlemci sayısını ve daha birçok kaynağı anlık olarak, ihtiyacına yönelik, artırıp azaltabilir.

1.3.1.3. Kullanıma Bağlı Ücretlendirme

Bulut bilişim platformları herhangi bir kurulum vb. maliyeti gerektirmez. Kullanılan bant genişliği, makina sayısı, depolama alanı gibi hizmet alan kişilerin kullanımına bağlı olarak bir ücretlendirme çıkarılır. Dolayısıyla ihtiyaç durumunda belli kriterler azaltılarak ekstra maliyetten kaçılabilir.

Sistem yöneticileri genelde donanım sağlama ve daha iyi altyapı kullanımı konusunda endişelenmektedir. Bulut teknolojileri ile kaynakları, kullandığın-kadar-öde felsefine göre uygulamaların ihtiyaçları oldukça kaynak istemesiyle daha etkili ve efektif yönetebilmektedirler (Zhao ve diğerleri, 2014:20).

1.3.1.4. Servis Düzeyi Anlaşması (Service Level Agreement – SLA)

Çoğu bulut bilişim platformu, verdikleri hizmetin sürekliliği ve güvenliğini garantieden Servis Düzeyi Anlaşması'na sahiptir (Krishnan, 2010:7). Bu sayede hizmette süreklilik ve yüksek erişilebilirlik sağlanmış olur. Sağlayıcı herhangi bir aksaklık durumunda müşterisine geri ödeme yapmayı garanti eder.

1.3.1.5. Dağıtılmış Veri Merkezleri

Bulut hizmet sağlayıcıları hizmetleri sunarken yapılarını dünya geneline dağıtmışlardır. Bahsedilen veri merkezleri büyük çaplı tesislerdir ve bakımı, güvenliği oldukça zordur. Bu nedenle sağlayıcılar very merkezlerini dünyadaki güvenli bölgelere kurmuşlardır. Bunun başlıca sebepleri veri güvenliği ve doğal afet ile gelebilecek kazalardır. Bu yapının oluşturulması; ulusal ya da uluslararası yasalar, jeopolitik fikirler, yük dengeleme, ağ gecikmelerini minimuma indirme gibi durumlar açısından gereklidir (Krishnan, 2010:7).

Birçok sağlayıcı hizmet sunarken bir merkezi ana merkez(master), diğer bir merkezide kopya(replica) merkezi olarak size sunar. Bazı sağlayıcılar ana merkez seçimini size bırakabilir ki bu veri iletişimi hızına etki eden bir faktördür. Hizmet sağlanayacağınız kullanıcılar hangi ülkede ise, veri merkezini bu ülkeye en yakın seçmek en mantıklısıdır.

1.3.1.6. Bakım

Bulut bilişimde alınan hizmete bağlı olarak herhangi bir kurulum ve bakım yapılması gerekmez. İlerleyen sayfalarda incelenecek olan PaaS ve SaaS hizmetlerinde, hizmeti satın alan hiçbir bakım ve kurulum işlemiyle uğraşmaz.

Bulut bilişimde, her bir sunucu üzerine ayrı ayrı kurulum yapılmamakta, tek bir noktada yer alan sistem görüntüsü, devreye giren tüm sunucular üzerine otomatik olarak kopyalanmaktadır. Bu sayede, herhangi bir güncelleme işleminde, temel sistem görüntüsü üzerinde yapılan değişiklikler, tüm cihazlara otomatik olarak yansıtılabilmektedir (Sevli, 2010:18).

1.3.1.7. Performans

Hizmet sağlayıcıları, sağlamış oldukları sunucu itibarıyla son teknoloji ürünler kullanmaktadır. Dolayısıyla hızlı bir işlem performansı sağlamaktadır. Ayrıca sağlayıcıların sunmuş olduğu performans izleme yazılımları yardımıyla hızınızı ölçebilir, ihtiyaca göre sunucu sayısı artırılabilir.

1.3.8. Bulut Bilişim Dezavantajları

Bulut bilişimde bazı dezavantajlar bulunmaktadır. Hizmeti yani donanımı bir sağlayıcıdan kullandığınızdan sağlayıcıya bağlı kalmaktasınızdır. Sağlayıcıda meydana gelecek herhangi bir aksaklık hizmetlerinizin durmasına yol açabilir. Çoğu sağlayıcı %99.9 oranında erişilebilirlik garantisi vermektedir. 2014 yılının yazında Microsoft Azure platformda meydana gelen aksaklıktan dolayı birçok firma hizmetlerinde duraksama yaşamıştır. Bu süre 54 saati bulmuş ve birçok firma ticari açıdan zarar uğramıştır. Aynı şekilde Google 2014 yılında 15 dakika süre boyunca durmuş, hizmetlerinde aksaklıklar yaşanmıştır.

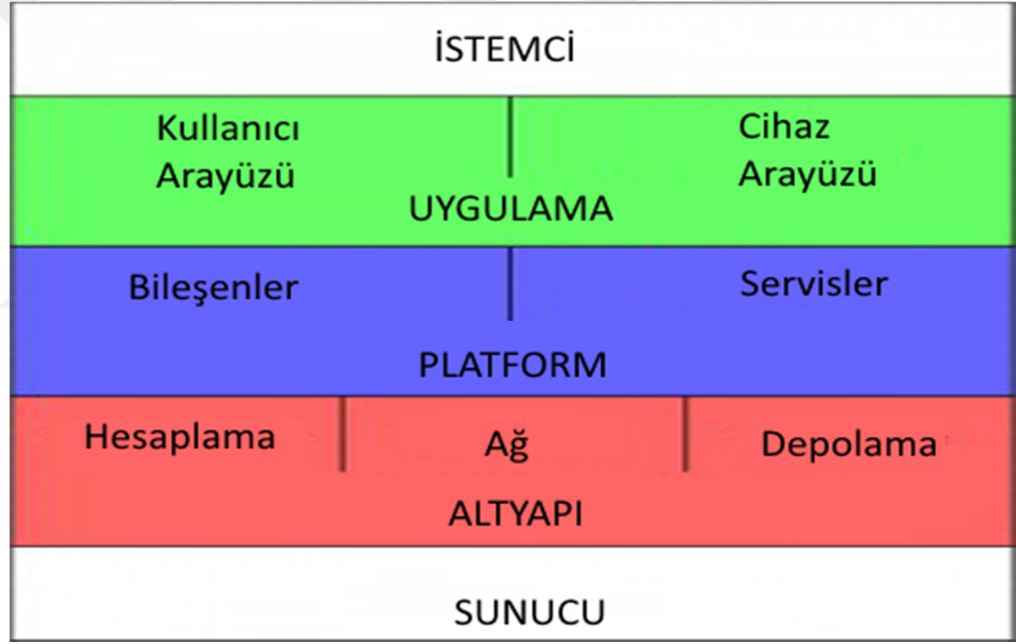
Bir başka dezavantaj olarak nitelendirilebilecek konu ise bulut bilişimin platformlarının sürekli, sabit ve hızlı bir internet bağlantısı gerektirmesidir. İnternet bağlantısının sekmeye uğraması, kullanıcıların hizmetlerde faydalanamamaları anlamına gelmektedir. Bu siber bir saldırı, donanım arızaları, internet sağlayıcılarındaki aksaklıklar gibi birçok sebepten ötürü olabilir. Bu aksaklıklara karşı hizmet sağlayıcıların önemlemlerini almaları gerekmektedir.

Veri güvenliği de ayrıca başka bir sorun olabilir. Bulut bilişimin yükselen bir trend haline gelmesiyle verinin çeşitli sistemler arasında paylaşılmasında bazı zayıflıklar ortaya çıkabilir(Chandra ve Angepat, 2010:3). Veriler sağlayıcıların veri merkezlerinde

olduğundan, bu verilerin güvenliğini sağlamak bulut sağlayıcısının sorumluluğundadır. Çok gizli verileri bulut platformlarında saklamamak sağlıklı bir karar olacaktır. Bu yüzden bankalar kendi platformlarından hizmet sağlamaktadırlar. İlla verileri bulut sağlayıcısında tutmak istiyorsanız, verilerini şifreleyip saklamanız sizin adınıza bir avantajdır.

1.4. Bulut Bilişim Katmanları

Bulut bilişim, bulut hesaplama çözümleri adı altında bazı katmanlarda oluşur. Bu katmanlar sunucu, altyapı, platform, uygulama ve istemci kısımlarıdır. Bu katmanların her birisi yapıda önemli rol oynamaktadır. Aşağıdaki figür bulut yapısını göstermektedir



Şekil 5: Bulut Bilişim Katmanları

Bu katmanların incelenmesi aşağıdaki gibidir:

1.4.1.İstemci

İstemciler birtakım donanım ve/veya yazılımlardan oluşan ve cloud'a bağlı olarak çalışan, cloud olmadığında herhangi bir işlevi olmayan elemanlardır. Örnek olarak bilgisayarlar, telefonlar, işletim sistemleri verilebilir

1.4.2.Uygulama

Uygulamaların internet üzerinden sunulduğu servis olarak düşünülebilir. Örnek olarak web sunucusu gösterilebilir.

1.4.3.Platform

Platform, bulut yazılımları ve işlem birimlerini, bulut mimarisi içerisinde yer alan kullanıcılara sunan servisleri barındırır. Bu kısım, firmaları donanım maliyetinden kurtarır, anlık ölçeklendirme avantajı sağlar.

1.4.4.Altyapı

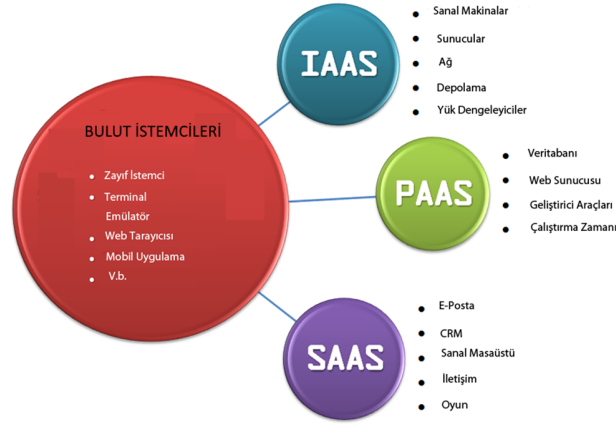
Altyapı kısmı bilgisayarları, depolama alanlarını, denge yükleyicilerini, ağ, güç üniteleri gibi fiziksel ve sanallaştırma gibi sanal kısımları barındıran bölümdür.

1.4.5.Sunucu

Sistemin en alt katmanındaki elemandır. Bulut sistemleri, çok sayıda sunucunun bir araya gelmesiyle oluşan devasa konteynerlardır. En temelde fiziksel bir sunucu gibi düşünülebilir.

1.5.Bulut Bilişim Servisleri

Aşağıdaki şekilde bulut bilişim servisleri gösterilmiştir.



Şekil 6: Bulut Bilişim Servisleri

Kaynak: (Tech Aditya,2015:1)

1.5.1. Altyapı Hizmeti (IaaS – Infrastructure as a Service)

Bu hizmet modelinde sağlayıcılar, sunucuları fiziksel ve sanal makine olarak sunmaktadırlar. IaaS, özgün IP adreslerine sahip sanal makineler ile talebe bağlı depolama alanları sağlamaktadır (Kulkarni, Sutar ve Gambhir, 2012:118). Sanal makineler Xen veya KVM gibi yazılımlar yardımıyla oluşturulurlar ve yönetilirler. IaaS, self servis hizmet modeli gibidir. Kullanıcılar görüntüleme, erişim, uzak altyapıları yönetme, depolama, ağ yapılandırması gibi işlemleri kendileri yaparlar. Yani donanım satın almak yerine donanımı kiralamış olurlar ve yönetimlerini kendileri yaparlar. IaaS'a örnek olarak AWS, Microsoft Azure, Google Compute Engine (GCE) gösterilebilir.

1.5.2. Platform Hizmeti (PaaS – Platform as a Service)

Bu modelde sağlayıcılar genelde işletim sistemi, programlama dili yürütme ortamı, veri tabanı, web sunucusu gibi hizmetler sunarlar. PaaS, istemcilere ön yapılandırılmış platform sunmaktadır. Arka planda uygulama gereksinimlerine göre altyapıyı otomatik olarak ölçeklendirmektedir (Gorelik, 2013:20). Kullanıcılar bu hizmet modeli ile uygulama geliştirme ve düzenleme yapabilmektedirler. PaaS, geliştirme, test ve uygulama yükleme işlemlerini oldukça kolaylaştırmakta ve maliyeti düşürmektedir. Bu hizmete Engine Yard örnek gösterilebilir. Engine Yard, sağlamış olduğu paneller ile size yönlendirmektedir. Yükleyeceğiniz uygulamanın hangi dilde olduğu, hangi

veritabanını kullanacağınızı, kaçtane instance çalıştıracağınız gibi parametreleri, sunmuş olduğu arayüzler yardımıyla seçtirir. Kullanıcıya kalan, ayağa kaldırmış olduğu sunucu kümelerine uygulamasını yüklemektir.

1.5.3. Yazılım Hizmeti (SaaS- Software as a Service)

Bu hizmet modelinde kullanıcı şirketler PaaS ya da IaaS kullanarak yüklemiş oldukları uygulamaları son kullanıcıya sunarlar. SaaS, çevrimiçi hazır yazılım çözümleridir (Gorelik 2013:20). Hemen hemen bütün SaaS uygulamalarına tarayıcı kullanarak erişilebilir, herhangi bir indirme ve kurulum gerektirmez. SaaS uygulamalarına örnek olarak Google Docs, Microsoft Office 365, Salesforce gösterilebilir.

1.6. Bulut Bilişim Yerleştirme Modelleri

Yerleştirme modelleri, bulut yapısının kullanımı ve barındırılmasıyla alakalı modellerdir. Bulut hizmetleri herkes tarafından kullanılacak şekilde ya da kullanıcılara özel yapılandırılmış olabilir. Bu yapılandırma modelleri şöyledir:

1.6.1. Özel Bulut

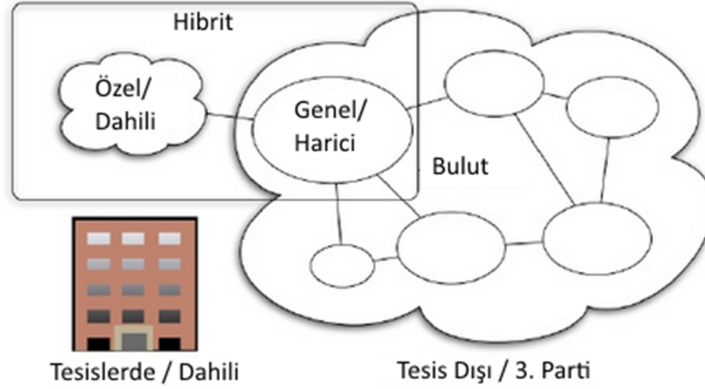
Özel bulut, sadece tek bir organizasyon için kullanılan bulut tipidir. Firma bünyesinde ve ya barındırma şirketlerinde barındırılması yapılabilir. Doğru yapıldığı zaman, iş üzerinde pozitif bir etkiye sahip olabilir, fakat yönetilmesi için ciddi emekler sarfetmek gerekmektedir, güvenlik önlemleri çok iyi alınmalıdır.

1.6.2. Genel Bulut

Genel bulut, internet üzerinden kullanıma sunulan ve ortak kullanıma açık buluttur. Örnek olarak AWS, Microsoft ve Google gösterilebilir.

1.6.3. Hibrit Bulut

Hibrit bulut, iki ya da daha fazla bulutun (özel ve genel) birleşmesinden oluşan bulut tipidir. Hibrit mimarisi hem şirket içi kaynaklara hem de sunucu tabanlı bulut yapısına ihtiyaç duyar.



Şekil 7: Bulut Bilişim Tipleri

Kaynak: (Salesforce Tutorial, 2013)

1.7. Bulut Bilişim Teknolojileri

Uluslararası Standart ve Teknoloji Enstitüsü'ne (NIST) göre bulut bilişim, uygun ve talebe bağlı ağ erişimi ile yapılandırılabilir bir kaynak havuzuna erişimi sağlayan, hızlıca karar alınabilen ve en az çaba sarfetmenizi sağlayan bir servis modelidir. Bulut bilişim aşağıdaki kategorilerde incelemek mümkündür:

1.7.1. Internet Teknolojileri

Web Servisleri: Web servisler cihazlar arasında ağ üzerinden bir haberleşme methodudur. W3C web servileri, uyumlu olmayan makineler arasında ağ üzerinden etkileşim method şeklinde tanımlanmaktadır. Bir web servis makine işlenebilir arayüzüne sahiptir (WSDL). Diğer sistemler web servis ile Basit Nesne Erişim Protokol'ünü (SOAP) kullanarak web servisin tanımlarını okur ve haberleşir.

Servise Dayalı Mimari (SOA) : bilgisayarda sistemlerin işlevselliklerini iş süreçleri etrafında gruplaştırarak sistem geliştirmesi ve bütünleştirilmesinde yol gösteren bir yazılım tasarımı felsefesidir. HYM altyapısı, iş süreçlerine katılan değişik yazılım uygulamalarının birbirleriyle haberleşmek üzere yazılmamış olmalarına rağmen veri alışverişinde bulunmalarını sağlar. Hizmet yönelimi, hizmetlerin işletim

sistemleri, programlama dilleri ve diğer teknolojik ayrıntılarla ancak gevşek bir bağ oluşturmasını hedeflenmektedir.

1.7.2. Dağıtık Hesaplama

Grid Hesaplama: Grid hesaplama, farklı lokasyonlardaki bilgisayar kaynaklarının bir amaca ulaşmak için bir araya getirilmesidir. Grid hesaplama zaman zaman bulut hesaplama ile karıştırılmasına rağmen farklı bir olaydır. Grid hesaplama, farklı domainlerdeki bilgisayarları bir araya getirir. Grid hesaplama ana stratejilerden birisi, bir ara katmanı bir programı birden fazla iş parçacığına bölmesi için kullanmasıdır. Bölünen bu iş parçacıkları yüzlerce, binlerce farklı bilgisayara hesaplanması için dağıtılır.

Utility (Yarar – Hizmet) Hesaplama: Hesaplama ve depolama kaynaklarının istemcilere hesaplanarak dağıtılması fikridir. Evlerde yer alan elektrik, su, doğalgaz kullanımı konseptindedir. Fikir 1960’larda ortaya çıkmıştır fakar donanım yetersizliğinden rafa kalkmıştır.

1.7.3. Kaynak Sanallaştırma

Sanallaştırma birçok istemciye hizmet verebilmek ve büyük veri merkezleri oluşturmak için geliştirilmiş bir yazılımdır. Ana fikir olarak sanallaştırma, donanım kaynaklarını istemciler arasında paylaşmayı hedef alır. Sanallaştırma bir donanım üzerinde birden fazla işletim sistemi çalıştırılmasını sağlar. Bir VMM (Virtual Machine Monitor – Sanal Makine Monitörü) denilen yapı kaynaklara erişir ve her bir istemcinin işletim sistemi için bir VM oluşturmaya izin vererek kaynakları ortak kullanmasını sağlar. Örnek VMM olarak dünyada VMware, Xen, KVM popüler olanlardır.

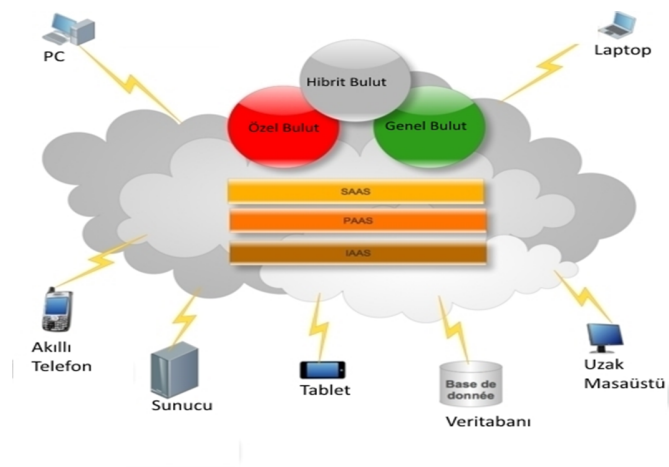
1.7.4. Otonomik Hesaplama

Otonomik hesaplama, dağıtım kaynakları kendi kendine yönetebilmesini hedefleyen bir hesaplama biçimidir. Yani kaynaklar kendini takip edebilmeli, kendini onarabilmeli ve optimize edebilmelidir. Otonomik hesaplama büyük veri merkezlerindeki bazı görevleri yerine getirebilmemize yardımcı olmuştur.

1.8. Bulut Bilişim Platformları

Bulut bilişim platformları, geliştiricilerin uygulamalarını yayınlıyabilecekleri altyapı hizmetleridir. Bulut bilişim yeniliği, depolama, sunucu ve uygulama gibi bilgisayar kaynaklarının oluşturduğu ağ havuzunu isteğe bağlı erişimi sağlamıştır (Nwobodo, 2015:283). Bulut platformları özellikle PaaS ile altyapı yönetim işlemlerini ortadan kaldırılıp, maliyeti azaltmaktadır. Bu yüzden geliştiriciler bulutun bu özelliğinden faydalanarak kolay bir şekilde uygulamalarını platformlarda yayınlabilir.

Her bulut platform sağlayıcısı birbirinden farklı hizmetler ve servisler sunmaktadır. Sağlayıcılar hizmetlerini soyutlayarak farklı programlama dillerine hizmet vermektedirler. Geliştiriciler platform seçmeden önce ihtiyaçlarını göz önünden bulundurmalı, hizmetlerden en iyi şekilde faydalanabilmek için platformları çok iyi incelemelidir.



Şekil 8: Bulut Bilişim Platformları

Kaynak: (Zoomdb24, 2015)

1.8.1. Amazon Web Servisleri (AWS)

AWS, uygulamaları ölçeklendirmek, bilgi teknolojileri masraflarını azaltmak ve organizasyonları hızlandırmak için yükleme servisleri, uygulama, depolama ve uzak hesaplama gibi bir çok hizmet sunan bir platformdur. Amazon.com şirketi tarafından

2006 yılında kurulmuştur ve 2007 yılında 180 bini aşkın geliştiricinin üye olduğu duyurulmuştur.

AWS ile dakikalar içerisinde hesaplama gücü, depolama alanı ve diğer servisler talep edebilir ve bir problemi çözmek için ihtiyacınız olan geliştirme ortamı veya programlama modelini seçme esnekliğine sahip olabilirsiniz (Varia ve Mathew, 2014:4).

Amazon Web Servisleri, kullanıcıları HTTP üzerinden REST ve SOAP protokolleri ile hizmet sunar. Bütün servisler kullanım bazlı faturalandırılır fakat kullanım hesaplamaları servisten servise değişkenlik göstermektedir.



Şekil 9: Amazon Web Servisleri Yönetim Paneli

Kaynak: (Amazon, 2016)

Amazon Web Servisleri bir çok hizmet sunmasına rağmen başlıca hizmetleri aşağıda incelenmiştir:

1.8.1.1. Amazon S3 (Amazon Simple Storage Service)

Amazon S3 geliştiricilere ve bilgi teknolojileri takımlarına güvenilir, dayanıklı, yüksek ölçeklenebilir nesne depolama hizmeti sağlar. Amazon S3 kullanımı oldukça kolaydır, internet üzerinden sade bir web servis ile veri okuma ve yazma işlemi yapılabilmektedir.

Nesneler, bucket (kova) adı verilen nesneler içerisinde saklanabilmektedir. Belirlenen bucket'lere erişim <http://s3.amazonaws.com/bucket/key> şeklinde bir adres yardımıyla olabilmektedir. 2013'te New York'ta yapılan AWS summit organizasyonunda Amazon CTO 'su Werner WOGELS, 2 trilyondan fazla nesnenin Amazon S3'te olduğunu söylemiştir.

Amazon S3, .Net, Java, Python, Ruby, PHP gibi birçok programlama diline destek vermektedir.

1.8.1.2. Amazon SQS (Amazon Simple Queue Service)

Amazon SQS, güvenilir, hızlı, ölçeklenebilir bir kuyruklama servsidir. Queue sistemi bilgisayar üzerinde çalışan ve ram'de depolama yapan, farklı bir programdan diğer programa veri iletişimi yapmanızı sağlayan, bunu yaparken programlarını birbirinin cevap beklemesini ortadan kaldırmanızı sağlayan bir yapıdır. Ram üzerinde çalıştığından okuma ve yazma işlemleri oldukça hızlıdır.

Amazon SQS ise queue sisteminizi bulut üzerinde kullanmanızı sağlar. Bir queue sistemi kurmanız için gereken 3. parti uygulama ve donanım maliyetini ve zaman kaybını azaltır.

1.8.1.3. Amazon RDS (Amazon Relational Database Service)

Amazon RDS bulut üzerinde ilişkili veri tabanı yürütmenizi sağlayan hizmettir. Karışık admin işlemlerini, vertiabanı kurulumu, yedekleme, felaket yönetimi gibi, otomatik olarak yapar ve size güvenli bir sistem sunar. Sağlımış olduğu konsol arayüzü yardımıyla yapılandırma kurulabilir.

Amazon RDS, Amazon Aurora, MariaDB, MYSQL, PostgreSQL, ORACLE ve MSSQL hizmetleri vermektedir. Fiyatlandır veritabanı depolama büyüklüğüne bağlı olarak değişkenlik göstermektedir.

1.8.1.4. Amazon E2 (Amazon Elastic Compute Cloud)

Amazon EC2, AWS'nin merkezinde bulunan, kullanıcılara sanal bilgisayar kiralama olanağı sağlayan hizmettir. Kullanıcılar bir web servis arayüzü ile bir bilgisayar image(görüntü) ayağa kaldırabilir ve sanal makine oluşturabilir.

Yapılandırmak olarak istediğiniz sayıda bilgisayar yapılandırabilirsiniz ve bu yapılandırma sayısı, bilgisayar özelliklerine bağlı olarak ücret ödersiniz. EC2, bir IaaS hizmetine güzel bir örnektir. Dolayısıyla yeni bir donanım alıp, bu donanımı yapılandırmak ile uğraşma zahmetinden kurtulmuş olursunuz. Daha sonra kullanmış olduğunuzu donanım ihtiyaçlarınız azaldığında, kiralama sayınızı azaltabilir ve dolayısıyla maliyetinizi de azaltabilirsiniz.

1.8.1.5. Amazon ElasticSearch Service

Amazon ElasticSearch Servisi, ElasticSearch uygulamasını yapılandırıp yönetebileceğiniz bir hizmet türüdür. ElasticSearch, açık kaynak kodlu olan ve analitik motoru olarak kullanılan bir uygulamadır. ElasticSearch ile gerçek zamanlı uygulama görüntüleme, log analitikleri yapılabilmektedir. ElasticSearch ilerleyen sayfalarda detaylı olarak incelenecektir.

Amazon ElasticSearch Servisi, bir arayüz ile sunucu kümeleri oluşturma, uygulama büyüklüğü seçme gibi yapılandırma işlemlerini sunar. Bu servisi kullanarak tüm yapılandırma zaman ve ücret maliyetlerinden tasarruf edilir. Ölçeklenebilir olduğundan zaman içerisinde kullanım ihtiyacına göre ElasticSearch kümeleri azaltılıp artırılabilir.

1.8.2. Google Cloud Hizmetleri

Google, bulut hizmeti olarak hızlı bir altyapı sunmaktadır. Google cloud platformu geliştiricilerin uygulama geliştirip web uygulamaları barındırabileceği bir grup hesaplama aracıdır (Forrest, 2014).

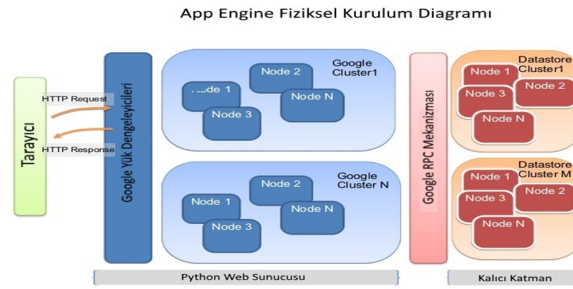
Sunduğu alt bulut hizmetlerini kullanarak altyapınızı oluşturabilir, milyarlarca arama kaydını milisaniyeler içerisinde yerine getirebilirsiniz. Youtube bu sayede 6 milyar saat video kaydı sunmakta, Gmail ise 425 milyon kullanıcıya bu alt yapı ile hizmet vermektedir.

Google ilk hizmetini 2008 yılında kullanıma sunmuştur. Günümüze kadar birçok hizmet devreye girmiştir. Hesaplama, depolama, ağ işlemleri, büyük veri, bazı API servisleri (çeviri, tahminleme gibi) bulut hizmetleri sunar. Aşağıdaki Google'ın sunmuş olduğu bazı hizmetler yer almaktadır:

1.8.2.1. Google App Engine

Google App Engine, ölçeklenebilir web uygulamaları ve mobil servisleri geliştirebileceğiniz bir platformdur. App Engine, dâhili servis API'leri ile NOSQL veri depoları, memcache, son kullanıcı yetkilendirme API gibi birçok uygulamada ortak olarak kullanılan hizmetleri sunar.

App Engine, uygulamalarını otomatik olarak ölçeklendirir. Bu ölçeklendirme sınırlarını sizin belirlediğiniz ve ücretini ödeyeceğiniz sınırlar içerisinde tutar. Bu bağlanda sunucu bakımı yapmanıza gerekmez.



Şekil 10: Google App Engine Uygulama Sunucu Yapısı

Kaynak: (Google, 2016)

Güncel olarak desteklenen programlama dilleri arasında Python, Java (diğer JVM dilleri yani Groovy, Jruby, Scala, Clojure), Go, PHP yer almaktadır. Google'ın açıklamasına

göre gelecekte dil bağımsız olacağı belirtilmiştir. Google % 99.95 uptime sağladığını belirtmektedir.

1.8.2.2. Google Compute Engine

Compute Engine, hizmet olarak sanal makine sağlamaktadır. Yani bu hizmet IaaS hizmetidir. Kullanıcılar istekleri doğrultusunda VM çalıştırabilmektedirler. VM'ler standart kopya ve ya bazı hazır hizmetleri barındıran kopyalar olarak ayağa kaldırılabilirler. Bu hizmete, sunulan geliştirici terminal, RESTful API ve ya Command Line arayüzü yardımıyla erişilebilmektedir.

Google bu hizmeti ilk olarak sınırlı modda 2012 de hizmete sunmuştur.

Aşağıdaki resimde AWS Server ve Google Compute Engine karşılaştırılması yapılmıştır:

	 AWS Server	 Google Server
Low Spec:	m1.medium	n1-standard-1
-- CPU	2 x 1 Ghz	3 x 1Ghz
-- RAM	3.75 GB	3.75 GB
-- HDD	410 GB	No attached storage
High Spec:	m3.2Xlarge	n1.standard-8
-- CPU	26 x 1Ghz	22 x 1Ghz
-- RAM	30 GB	30 GB
-- HDD	No attached storage	No attached storage

Şekil 11: Google Compute Engine Ve Amazon Web Servis Karşılaştırması

Kaynak: (Amazon, 2016)

1.8.2.3. Google Cloud Storage

Google Cloud Storege, RESTful servisleri ile sunulan dosya depolama hizmetidir. Hizmet Google bulut hizmetlerinin performans ve ölçeklenebilirliğini sağladığı gibi, güvenilirdir.

STANDART, DRA ve NEARLINE şeklinde 3 tipte sunulmaktadır. Tabi ki bu 3 hizmetin hız ve ücretlendirmeleri birbirinden farklıdır.

Cloud Storage, Amazon S3 gibi dosyaları bucket denilen yapılarda tutar. Yaklaşık olarak 107 GB limitli dosya yüklenmesine izin vermektedir.

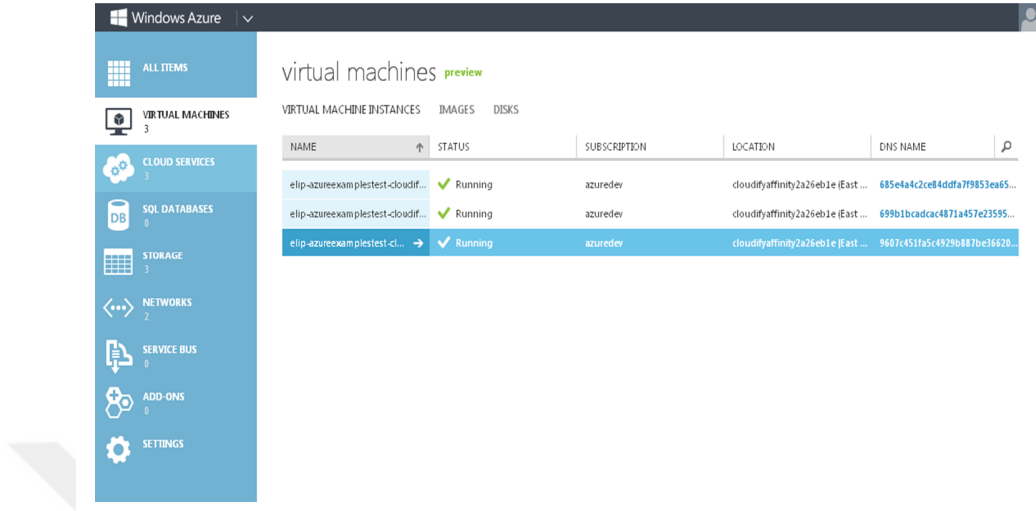
1.8.3. Microsoft Azure Platforms

Microsoft Azure, bulut bilişim platformu ve altyapısıdır. İlk olarak 2010 yılında Windows Azure adıyla çıkmıştır. Azure Paas ve IaaS hizmetleri sağlamak ve birçok programa dilini desteklemektedir. Uygulamalar için bir çalıştırma olarak hizmet veren Windows Azure adlı bir bulut işletim sistemini sağlamaktadır. Microsoft Azure servis platformları, uygulama geliştiricilerine her biri ayrı ayrı bir servis sunan bir grup bulut teknolojisidir (Chappel, 2008:3).

Amazon Web Servisleri gibi Azure da ihtiyaca yönelik ölçeklendirme sunmaktadır. Ölçeğinizi artırabilir, artırdığınızı ve ya azalttığınızı ölçüde ücretlendirilirsiniz.

Azure birçok programla dili ve windows işletim sistemlerinin yanı sıra, birçok linux tabanlı, ubuntu, suse ve debian gibi versiyonları da desteklemektedir.

C# kullanarak geliştirdiğiniz uygulamalar için Visual Studio Azure entegrasyonu bulunmaktadır ve bu sayede geliştirdiğiniz uygulamaları kolayca Azure üzerinde yayınlatabilirsiniz.



Şekil 12: Microsoft Azure Platform Ana Panel

Kaynak: (Microsoft Azure, 2013)

1.8.3.1. Microsoft Azure App Servisleri

Azure App Servisleri bir PaaS tipinde bir servis olup geliştiricilere uygulamaları barındırabilecekleri ve yönetebilecekleri hizmeti sunar. Bu serviste 2 tip uygulama barındırabilirsiniz. Bunlar Websites ve Webjobs uygulamalarıdır.

Websites ile Microsoft web programlama dilleri, nodejs, php ve python gibi web uygulamaları barındırıp hizmet verebilirsiniz. Azure GIT, FTP, TFS gibi aracı programlar ile platformuna yüklemenize izin vermektedir.

Webjobs, mobil uygulamalar gibi uygulamalar için arka plan servisleri sunmanıza olanak sağlar.

1.8.3.2. Microsoft Azure Big Compute

Azure talebe bağlı olarak büyük paralel hesaplama işlemleri için hesaplama kaynakları sağlamaktadır. Bu tarz hesaplama işlemleri için büyük bilgisayar kümeleri kirayalabilmemize olanak verir. Büyük hesaplama işlemlerinden kasıt günler süren hesaplama işlemleridir. Bu işlemleri yapmak tek bir sunucuda çok zordur. Örnek olarak

Netflix pazarlama departmanında kullanılan ve kullanıcı verilerine göre analiz yapıp çıkarımlarda bulunan scriptler gösterilebilir.

1.8.3.3. Microsoft Azure Mobile Services

Azure Mobile Servisleri istemci ve mobil uygulamalarınız için hızlı, ölçeklenebilir bir bulut arkaplan servis hizmeti sağlar. Bulut üzerinde veri saklamanıza olanak sağlayarak kullanıcı ve cihazları ayırmanıza, bu iki unsuru entegre etmek için bir yetkilendirme koymanıza olanak oluşturur. Örnek olarak mobil uygulamalara “push notification” yani bildirim gönderme uygulaması gösterilebilir.

Mobil servisler ile veri alışverişi RESTful servisler ile JSON formatında sağlanır.

1.8.3.4. Microsoft Azure Storage Services

Azure storage, büyük boyutlarda verileri kaydetmek ve ihtiyacınız olduğunda bu verileri okumanızı sağlayacak esnek servisleri sağlar. Azure storage, oldukça stabil ve yüksek erişilebilir özelliği ile hesap başına 500 TB ‘a kadar boyutta veri saklamanızı sağlar. Yüksek ölçeklenebilir yapısı ile kullandığınız alan kadar ödeme yapmanıza olanak sağlar.

Azure storage sunmuş olduğu REST API servisleri ile kolayca veri yüklenmenizi ve okumanızı sağlar. Bunun yanı sıra .NET, Java, C++ Node.js gibi birçok güncel popüler programlama dili için kütüphaneleri mevcuttur.

Azure storage 4 çeşit hizmet sunar ve bunlar Blob, Table, Queue ve File Storage’dır.

Blob ile dosya barındırma işlemleri yapabilirsiniz. Blob herhangi bir dosya olabilir ve byte veri şeklinde saklanabilir.

Table ile yapılandırılmış veri kümeleri depolanabilir. Table storage key-attribute (anahtar –özellik) yapısındadır ve veri kümelerine hızlı erişim sağlar.

Queue ile güvenilir mesajlaşma akışı oluşturulabilir. Queue, uygulamalar arası veri aktarımı için oldukça efektifir.

File Storage SMB protokolü kullanan uygulamalar için paylaşılan depolama alanı sunar.

1.8.3.5. Microsoft Azure Data Management

Azure Data Management (Azure Veri Yönetimi) veritabanı bağlamında hizmetler sunmaktadır. Bu hizmetlerden bazıları şöyledir:

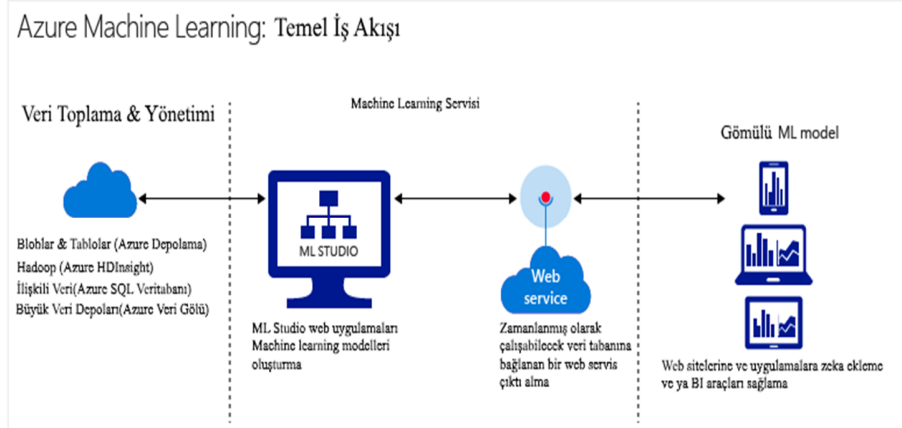
SQL Database, SQL Azure Database olarak bilinir ve Microsoft SQL Server kullanarak bulut üzerinde geniş, ölçeklenebilir, ilişkili veritabanları oluşturmanızı sağlar.

Azure Search, Tam-Metin aramaları yapılmasını sağlayan servisi sunar. Bu bağlamda ilerleyen noktalarda incelenecek olan ElasticSearch'e benzemektedir. REST Api'leri sayesinde JSON formatında veri alışverişi yapılabilmektedir.

Document DB, JSON formatında veri ile çalışan bir NoSQL veritabanıdır. Büyük ölçüde esneklik, ölçeklenebilirlik ve yüksek erişim sağlar. REST Api'ler yardımıyla veri alışverişi yapılabildiği gibi birçok dil için SDK'ları bulunmaktadır. Bu dillere örnek olarak .NET, Node.js, Java ve Python gösterilebilir.

1.8.3.6. Microsoft Azure Machine Learning

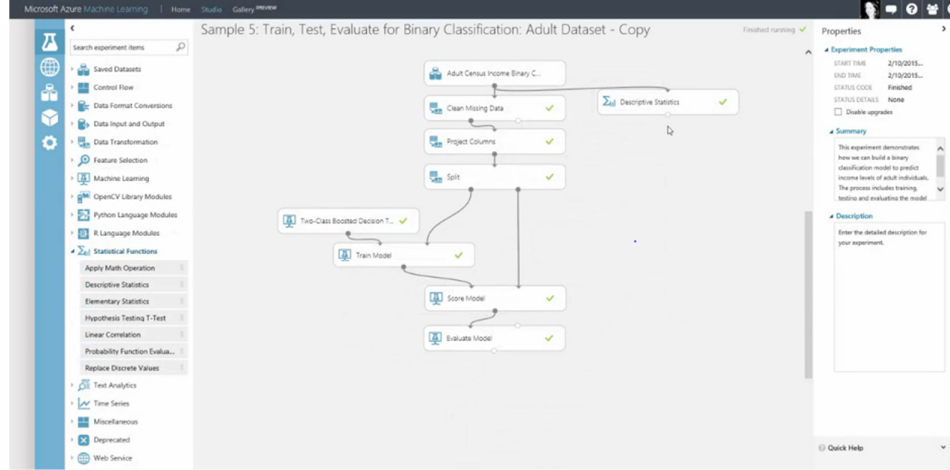
Machine Learning, yani Makine Öğrenmesi, bilgisayarlar üzerinde çıkarımlar yapmak için modeller ve küçük yazılımlar çalıştıran ve var olan verilerden çıkarım yapmamızı sağlayan konsepttir ve azure bulut yapısını kullanarak alaniz yapılmasını sağlar.



Şekil 13: Azure Makine Öğrenmesi İş Akışı

Kaynak: (Microsoft Azure, 2016)

Azure bu servisinde web üzerinde model tasarımları yapabileceğiniz bir uygulama sunar. Herhangi 3.parti bir uygulama indirip kurmanıza gerek yoktur. Aşağıdaki resimde modelleme gösterilmektedir;



Şekil 14: Azure Makine Öğrenmesi İş Akış Tasarımı

Kaynak: (Microsoft Azure, 2016)

Stüdyo ortamında tasarlamış olduğunuz modelleri, çalıştırılabilir web servislerine dönüştürmek mümkündür. Bu sayede analiz sonuçlarını diğer uygulamalar ile paylaşma imkânı vardır.

Model senaryoları R ve Python dillerini kullanarak oluşturmak ve verileri analiz etmek mümkündür.

1.8.4. Diğer Bulut Hizmetleri

Dünya genelinde bulunan daha birçok bulut hizmeti ve uygulaması bulunmaktadır. Bunlardan bazıları Tablo 1’de gösterilmiştir.

Tablo 1
Diğer Bulut Hizmetleri

Bulut Adı	Açıklama
Apple iCloud	iCloud, Apple ürünlerinizi birbirine bağlayan ve bu cihazlarınız arası verilerinizi paylaşmanızı sağlayan SaaS tipi bir cloud hizmetidir. İlk olarak 2011’de yayınlanmıştır ve 2013’te 320 milyon kullanıcıya ulaşmıştır.
Appscale	Appscale, Google App Engine uygulamalarını diğer cloud platformlar üzerinde çalıştırabilmenizi sağlayan açık kaynak kodlu bulut platformudur. Python, Go, PHP ve Java programlama dillerini desteklemektedir.
Caspio	Caspio temelde, iş uygulamalarınızı kodlama yapmadan geliştirebileceğiniz bir bulut platformudur. 2000 yılında başlayan Caspio, 2001 yılında ilk versiyon şeklinde yayınlanmıştır.
Heroku	Heroku, PaaS özellikte olan bir bulut bilişim platformudur. 2007 yılında ortaya çıkmıştır ve 2010 yılında Salesforce.com tarafından satın alınmıştır. Ruby, Java, Node.js, Scala, Clojure, Python, PHP ve Go dillerini desteklemektedir.
Engine Yard	Engine Yard, PaaS platform olaran Ruby on Rails, PHP ve Node.js uygulamalarınızı yükleyim çalıştırabileceğiniz bir platformdur. 2009 yılında hizmete girmiştir.

Force.com	Force.com bir PaaS bulut platformudur ve geliştiricilere Salesforce.com uygulamalarıyla entegreli çalışabilecek add-on uygulamalar geliştirmesine olanak sağlar.
Rackspace	Rackspace, özel, genel ve hibrit bulut hizmeti sunan bir bulut platform sağlayıcısıdır. Aynı zamanda özel sunucular da sunmaktadır. Rackspace, bilgi teknolojileri masraflarını azaltmak ve performansı artırmak amacıyla uygulama, depolama gibi bulut hizmetleri sunmaktadır.
OrangeScape	OrangeScape, bir Hindistan firması olan OrangeScape tarafından geliştirilmiş, PaaS bulut platformudur. İlk olarak 2008 yılında kullanıma sunulmuştur.
Sun Cloud	Sun Cloud, Sun Microsystems tarafından geliştirilmiş, web arayüzü ile işlem kaynaklarına erişim sağlanabilen bir bulut bilişim platformudur. İlk olarak Sun Grid olarak 2006 yılının Mart ayında ortaya çıkmıştır.
HP Cloud	HP Cloud, HP tarafından sunulan genel, özel, hibrit bulut servisi. İlk olarak beta sürümü 2012 yılında kullanıma sunulmuştur.
DigitalOcean Cloud	DigitalOcean, 2011 yılında kurulmuş bulut hizmeti sağlayan şirkettir. DigitalOcean, PaaS olarak hizmet vermektedir. Basit arayüzü ile “Droplet” adı verilen yani ihtiyaca göre değişik özelliklere sanal bir sunucu oluşturulabilir.

1.9. Bulut Bilişim Uygulamaları

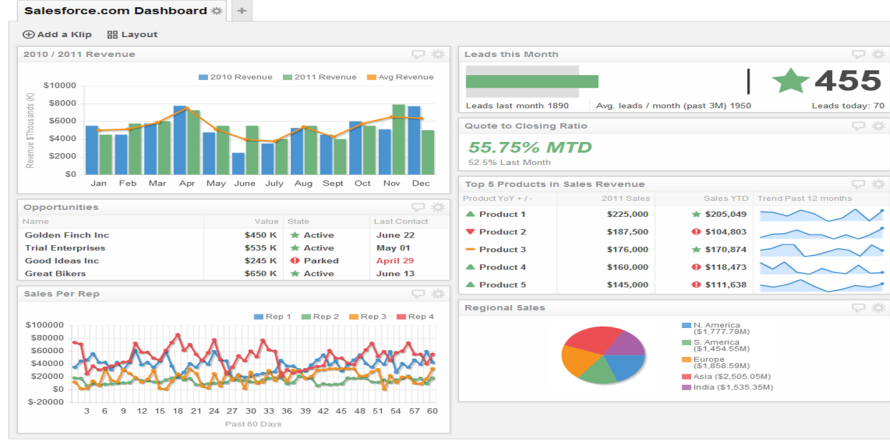
Bulut bilişim uygulamaları SaaS tipinde olan uygulamalardır. Çeşitli üreticiler, çeşitli sektörlerin ve kullanıcıların ihtiyacına yönelik geliştirmiş oldukları uygulamaları, bulut teknolojisinin yardımıyla kullanıcılara sunarlar. Günümüzde bulut altyapısını kullanarak geliştirilmiş birçok uygulama bulunmaktadır.

1.9.1. SalesForce.com

SalesForce, Amerikada kurulmuş bulut tabanlı çözümler üreten bir firmadır. 1999 yılında kurulmuştur ve SaaS uygulamaları geliştirmeyi hedef almıştır. Temel olarak

CRM servisi sunmaktadır. Yani bulut bilişimin gücünden faydalanarak, kullanıcılara browser üzerinden CRM konseptindeki hizmetleri sunar. Uygulama ile iş süreçlerinizi takip edebilir, kampanyalar düzenleyebilir, müşterilerinizin ihtiyaçlarını analiz edebilir, hesaplarınızı takip edebilir ve daha birçok CRM kapsamında işlemler yapabilirsiniz. Aynı işler akıllı mobil cihazlar üzerinde yapılabilir.

Altyapı olarak hiçbir şey ile ilgilenmeniz gerekmez. Salesforce, birçok ünlü bulut platformu sunan firmalardan güç almaktadır.



Şekil 15: Salesforce Panel

Kaynak: (Salesforce, 2016)

1.9.2. Kissflow.com

Kissflow, Orangescape şirketi tarafından geliştirilmiş ve 2012 de kullanıma sunulmuştur. Kissflow Google uygulamaları için bir iş akışı uygulamasıdır. Kissflow ile satın alma isteği, zamanlama onaylamaları, işe alın uyum süreçleri gibi birçok iş akışı süreci tasarlayabilirsiniz. Uygulama kullanıma başladıktan bu yana 10 bini aşkın kullanıcıya hizmet vermektedir. Klasik olarak bir süreç onay mekanizması yaratılabilmektedir. Her bir adımda farklı işlemler yapılabilmesine olanak sağlamaktadır.

Google uygulamaları entegrasyonu kısmında örnek olarak bir kullanıcı herhangi bir dosyayı kissflow süreci içerisinde ekleyebilir ve bu dosyayı Google domain'inde olan birisi ile paylaşabilmektedir. Uygulama arayüz olarak Gmail'e benzemektedir.

1.9.3. Google Docs

Google Docs, internet tarayıcısı üzerinden ofis uygulamalarını kullanabileceğiniz SaaS tipinde bir hizmettir. Google tarafından sunulmaya 2007 yılında başlanmıştır. Temel olarak bilgisayarınızda kullandığınız word, excel, powerpoint gibi ofis uygulamalarınızı, kendi sunmuş olduğu arayüz yardımıyla internet tarayıcısı üzerinden kullanabilmenizi sağlar. Bu anlamda platform bağımsız bir uygulama olduğundan oldukça efektifdir.

Ofiste oluşturmuş olduğunuz dosyaları direk olarak Google Docs'a yüklemeniz mümkündür. Bunun yanı sıra ihtiyaca göre anket formları düzenlenebilmekte, sonuçlar ile ilgili istatistik alınabilmektedir.

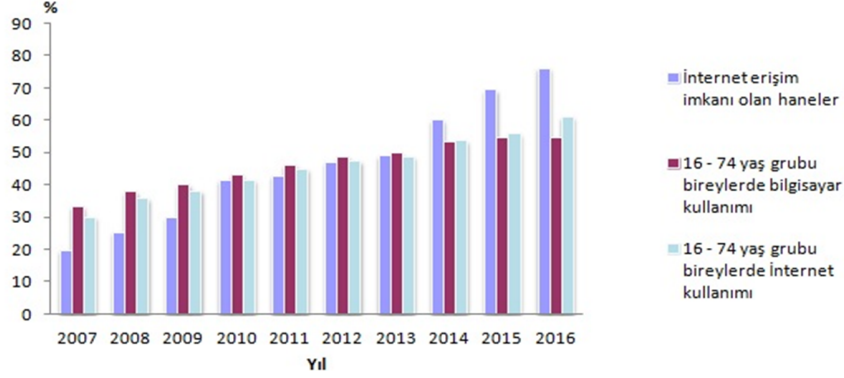
Aynı şekilde akıllı cihazlar üzerinden de oluşturulan dosyalara erişilebilmektedir.

1.10. Türkiye'de Bulut Bilişim

Dünyada olduğu gibi ülkemizde de bulut bilişim hizmetlerinden faydanılmakta, bulut bilişim alt yapısı kullanılarak hizmetler sunulmaktadır. Bulut bilişim hizmetleri kullanımı gün geçtikçe ülkemizde de artmaktadır. IDC tarafından Türkiye'de 30 BİT şirketine yapılan anketin analizlerine göre 2011 yılında bulut bilişim hizmetlerinin verilmesine yönelik yapılan toplam pazar harcamalarının 27,78 milyon ABD Dolarına ulaştığı tahmin edilmekle birlikte 2012-2016 döneminde ise yıllık %49,3 artışla 200 milyon ABD dolarını aşması beklenilmektedir. Söz konusu büyüklükte genel bulut pazarının 2016 yılında 150 milyon ABD Dolarının üzerinde olması hedeflenmektedir (IDC ve BTK, 2013).

TÜİK verilerine göre ülkemizde bilgisayar ve internet kullanım oranları 2016 yılı Nisan ayında 16-74 yaş grubundaki bireylerde sırasıyla %54,9 ve %61,2 olmuştur. Bu oranlar erkeklerde %64,1 ve %70,5 iken, kadınlarda %45,9 ve %51,9'dur. Bilgisayar ve internet

kullanım oranları, 2015 yılında sırasıyla %54,8 ve %55,9 civarında bulunmaktaydı. Temel göstergeler 2007-2016 arası Şekil 16'de gösterilmiştir.



Şekil 16: Hanehalkı Bilişim Teknolojileri Bilgisayar Kullanımı Temel Gösterler 2007-2016

Kaynak: (TÜİK, 2016:1)

Ülkemizdeki internet kullanımındaki bu artış, veriye olan talebi daha da artırmaktadır. Bu talep nedeniyle firmalar, uygulamadaki verileri kullanıcılara daha hızlı ulaştırmak için bulut çözümlerine yönelmektedirler.

Bulut bilişimin kamusal alandaki gelişmeleri, 4 Ekim 1983 tarih ve 77 sayılı Kanun Hükmünde Kararname (KHK) ile kurulan Bilim ve Teknoloji Yüksek Kurulu (BTYK)'nın 25. Toplantısında alınan kararla ivme kazanmaya başlamıştır. Türkiye Kamu Entegre Veri Merkezinin (TKEVM) kurulmasına ilişkin BTYK tarafından alınan kararın amacı; tüm kamu kurumlarının veri merkezlerinin birleştirilmesi sonucu tek bir veri merkezi kurulmasıdır (Bilgi Teknolojileri İletişim Kurumu, 2013:19).

Bu kapsamda;

- Ülkemizde kurumların e-Devlet hizmetlerini verebilmek için kendi altyapılarını geliştirmekte olması,
- Dünyadaki örnekler incelendiğinde veri merkezlerinin birleştirilmesine dair eğilimlerin gözlemlenmesi,

- Güney Kore'nin 48 merkezi kamu idaresinin bilgi sistemleri 2 ayrı şehre konumlandırılacak şekilde tek bir veri merkezinde birleştirilmiştir. Bu sayede 20 yedeklilik, felaket kurtarma merkezi, siber güvenlik, iş sürekliliği, kamu bulutu, etkin işletme maliyeti, kurumlar arası veri paylaşımı gibi hususların tamamına çözüm sağlanması,
- ABD, bulut bilişim ve ortak veri merkezi yaklaşımlarıyla 2015 yılına kadar 800'den fazla veri merkezini kapatmayı planlamaktadır. Bu kapsamda 2015 yılı sonunda 3 Milyar ABD Doları tasarruf edilmesinin öngörülmesi,
- İdari ihtiyaçlar, tasarruf imkânı ve siber güvenlik gereksinimleri doğrultusunda, halen her kurumda müstakil olarak işletilmekte olan veri merkezlerinin tek bir çatı altında birleştirilerek Türkiye Kamu Entegre Veri Merkezi'nin kurulması önem arz etmesi

Nedenleriyle BTYK'da "Kamu kurumlarının veri merkezlerinin birleştirilmesine yönelik hukuki, teknik ve idari yapılanma modelinin oluşturulmasına ve Türkiye Kamu Entegre Veri Merkezi'nin kurulması çalışmalarının yapılmasına" karar verilmiştir. Söz konusu kararın uygulanmasında sorumlu kuruluş olarak Ulaştırma, Denizcilik ve Haberleşme Bakanlığı görevlendirilmiştir. Kararla ilgili kuruluşlar da; Başbakanlık, Kalkınma Bakanlığı, TÜBİTAK ve TÜRKSAT olarak belirlenmiştir (Bilgi Teknolojileri İletişim Kurumu, 2013:19-20).

Ülkemizde birçok firma barındırma(hosting) hizmeti vermektedir. Bu firmalar sunmuş oldukları altyapıları belli bir yapıda barındırmaktadır. Bu sunucuların yapıları bir bakıma özel bulut olarak adlandırılabilir. Birden fazla istemciye hizmet verebilmek adına bu sunucular sanallaştırılmaktadır. Ülkemizdeki bulut hizmeti sunan firmaların altyapısı dünyadaki örnekleri ile karşılaştırıldıklarında daha basit seviyede kalmaktadır. Örneğin Amazon, Azure vb. gibi bulut hizmeti veren firmalar anlık olarak istemcinin sunucu, ağ trafiği, depolama alanı gibi hizmet değişikliklerini karşılayabilmektedir ve ücretlendirme kullanılan hizmet üzerinden hesaplanmaktadır. Sundukları yönetim panelleri sayesinde istemci isteklerine göre sunucular programatik bir şekilde devreye

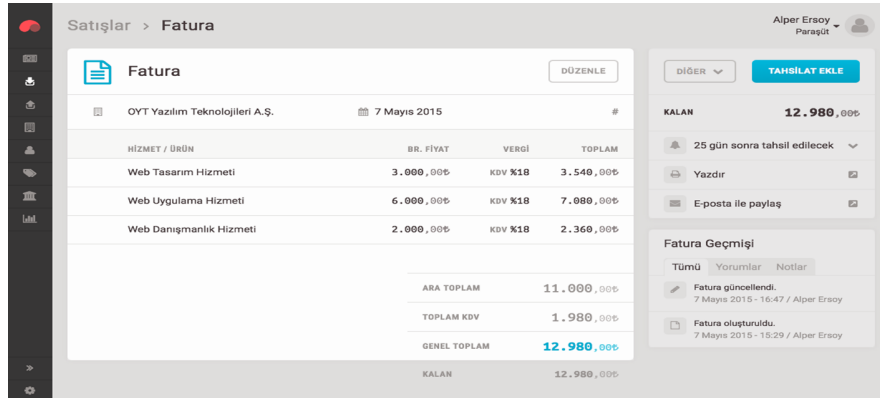
alınabilmektedir. Ülkemizdeki firmalar daha çok sabit özellikle altyapı sunmakta, anlık olarak değişim yapılmasına olanak sağlamamaktadır.

Ülkemizde bulut hizmetleri daha çok SaaS modelinde ilerlemektedir. Firmalar, bulut hizmeti veren dünyadaki büyük firmaların altyapısını kullanarak uygulamalarını bulutun gücüyle artırmaktadır. Son kullanıcıya hitap eden ya da altyapı olarak bulut hizmetlerinde faydalanan son kullanıcıya dolaylı hitap eden uygulamalar gün geçtikçe artmaktadır. Birçok firma veritabanı, depolama, önbellek işlemleri, uygulama barındırma, mobil servisler için bulut firmalarından yararlanmaktadır.

Ülkemizde yer alan bazı bulut çözümleri aşağıda incelenmiştir.

1.10.1. Paraşüt.com

Paraşüt.com, 2013 yılından bu yana bulut tabanlı finans ve fatura yönetim hizmetleri sunmaktadır. Yapılanbilen işlemler arasında fatura takibi, cari hesap takibi, gider takibi, raporlama, elektronik faturalama, çevrimiçi tahsilat gibi işlemler yer almaktadır. Herhangi bir kurulum yapmadan, aylık ücret bazında ücretlendirilerek internet tarayıcısı üzerinden kullanılabilir.



HİZMET / ÜRÜN	BR. FİYAT	VERGİ	TOPLAM
Web Tasarım Hizmeti	3.000,00	KDV %18	3.540,00
Web Uygulama Hizmeti	6.000,00	KDV %18	7.080,00
Web Danışmanlık Hizmeti	2.000,00	KDV %18	2.360,00
ARA TOPLAM			11.000,00
TOPLAM KDV			1.980,00
GENEL TOPLAM			12.980,00
KALAN			12.980,00

Şekil 17: Paraşüt Panel

Kaynak: (Tekzan, 2016)

1.10.2. Türk Telekom Bulut Ürünleri

1985 yılında kurulmuş ve Türkiye’de telekomünikasyon alanında liderlerden biri olan Türk Telekom, Amazon Web Servisleri altyapısını kullanarak bulut hizmetleri de vermeye başlamıştır. IaaS ve SaaS formlarında hizmetleri bulunmaktadır.

İhtiyaca göre, sunulan belli donanım özelliklerinden Linux ve Windows işletim sistemlerine sahip sunucuları kiralamak mümkündür. Bunun yanı sıra depolama uygulamaları, konferans uygulamaları, toplantı organize edip video konferansları yapılabilen uygulamalar, akademik olarak uzaktan eğitim ve sanal eğitim uygulamaları gibi birçok bulut tabanlı uygulamadan yararlanmak mümkündür.

1.10.3. Turkcell Akıllı Bulut

Turkcell akıllı bulut Turkcell bulut bilişim teknolojilerinin genel adı olarak adlandırılmaktadır. Turkcell akıllı bulut yapısına servis olarak altyapı, platform ve yazılım hizmetlerinin tümü dâhil edilmektedir. Turkcell akıllı bulut; sanal sunucu, kurumsal e-posta, akıllı faks, uzaktan yedekleme, akıllı dükkân paketi, yönetilebilir güvenlik hizmetleri, akıllı depo kurumsal gibi hizmetleri barındırmaktadır. 2007 yılından önce Turkcell içinde, ardından grup şirketlerine hizmet sunmaya başlamıştır. 2013 yılından itibaren de Turkcell akıllı bulut adı altında kurumsal hizmetler sunmaya başlamıştır.

1.10.4. Netmera

Netmera bulut platformu Inomera Research tarafından 2009 yılından itibaren Tübitak ve Kosgeb desteğiyle geliştirilmeye başlanmıştır. Platform ile daha çok mobil internet ve telekom sektörüne yönelik çözümler sunulmaktadır.

Netmera platformu bir Paas olup mobil uygulama geliştiricileri sunulan API, SDK ve back-end araç ve servisleri kullanarak kendi uygulamalarını kolaylıkla bulut bilişim altyapısı üzerine taşıyabilmekte veya mevcut uygulamalarına bulut üzerinde esnek biçimde büyüyen bir back-end ekleyebilmektedirler.

Netmera Amazon EC2 altyapısı üzerinde çalışmaktadır ve Java teknolojileri ile geliştirilmiştir. Netmera platformu pek çok servisi kolay kullanılabilir ve hazır olarak geliştiricilere sunmaktadır. Örneğin Netmera platformunda bir arama motoru Türkçe dilinde metin araması imkânı sunmakta, medya/fotoğraf tabanlı uygulamalar için API ve CDN hizmeti sunulmakta, lokasyon bazlı uygulamalar için kolaylaştırıcı araçlar ve lokasyon bazlı arama imkanı sunulmaktadır. Bu servisler ile içerik sunumuna dayanan veya lokasyon bazlı uygulamalar kolaylıkla geliştirilebilmektedir. Netmera çalışma yapısı Şekil 18’de gösterilmiştir.



Şekil 18: Netmera Yapısı

Kaynak: (SlideShare, 2016)

1.10.5. PlusClouds

PlusClouds H2B internet teknolojileri tarafından 2011 yılında geliştirilmeye başlanmış, temelde IaaS olmak üzere bu hizmet üzerinde kurulmuş çeşitli türev hizmetler de vermektedir. PlusClouds, sunucuları sanallaştırıp, bunlardan tam otomatik ve ihtiyaca yönelik çalışabilen sistemler kurmayı ve üçüncü parti çözüm ortaklarıyla beraber çeşitli çözümler sunmayı amaçlamaktadır.

IaaS'nin olmazsa olmazı dinamik olarak genişleyebilen veya daralabilen sunucular sağlamaktadır. Bunun yanında Cloud DNS çözümünü sınırlı domain için ücretsiz vermektedir. Platform üzerinde isteğe bağlı olarak içerik yönetimi, CRM ve ERP, iş

reklam yönetimi, veri depolama ve yedekleme, güvenlik ve performans, VoIP telefon santralleri kategorilerinde yazılım çözümleri sunmaktadır.



BÖLÜM 2: NOSQL (NON SQL, NOT ONLY SQL, NON RELATIONAL SQL)

2.1. NoSQL Nedir?

İlişkisel veritabanları 1970'lerden bu yana geliştirilmekte ve birçok uygulama için veri depolama sistemleri olarak kullanılmaktadır. İlk olarak IBM tarafından geliştirildiği bilinmektedir. İlişkisel veri tabanları temel olarak verileri belli bir yapıda saklayıp bu veriler üzerinde işlem yapabilmemizi sağlarlar. Yapının ilişkisel olarak düzgünlüğü, veriler üzerinde yapılacak işlemlerin performansını etkilemektedir. İlişkisel veritabanında veri saklayan bir uygulama, genel olarak veritabanının yapı bütünlüğüne göre hareket etmek zorunda kalır. Çalışma prensibi genel anlamda veritabanının tablo yapısına göre olmaktadır.

Bu yapı bütünlüğü güvenilirlik sağlarken bazı dezavantajları da beraberinde getirmektedir. NoSQL kavramı da bu noktada olan ihtiyaçtan doğmuş bir kavramdır.

NoSQL ilk olarak 1998 yılında İtalyan Carlo Strozzi tarafından geliştirilmiştir. Strozzi ürünü, veriyi SQL ile yapılandırmayı bıraktıktan sonra NoSQL olarak tanımlamaya başlamıştır.

Last. fm'den Johan Oskarsson 2009 yılında bir buluşma düzenleyerek NoSQL kavramını tekrar gündeme getirmiştir. Bu süreçten sonra birçok büyük firma, Google, Amazon gibi, NoSQL veritabanları geliştirmeye ve yayınlamaya devam etmiş olup, NoSQL hizmet sayılarında artış gözlemlenmeye başlamıştır.

İlişkili veritabanları, verilerin ilişkili bir modelde düzenlenmesinden oluşmaktadır (Codd, 1970:377). İlişkisel veri tabanı kullanarak veri depolama işlemleri bazen en iyi çözüm olmamaktadır. Yazılım geliştiriciler zaman zaman ilişkisel veri yapıları ile bellek veri yapıları arasındaki uyumsuzlıklardan dolayı sıkıntılar yaşamaktadırlar. NoSQL yazılımcılara hafıza veri yapılarını ilişkisel veri yapılarına dönüştürmeden geliştirme yapmalarını sağlamaktadır.

NoSQL'de LIKE benzeri SQL komutları olmadığından, veri okuma ve işleme hızı daha yüksektir. Bu performans, uygulamaların etkisini ve kullanıcı memnuniyetini artırmaktadır.

Klasik ilişkisel veritabanlarının getirmiş olduğu bir diğer sorun ise homojen data gerektirmesidir. Bunun anlamı, yapılandırılan şemada herhangi bir değişiklik olduğunda tablo düzeltmeleri yapılmak zorunda kalınmadır. Bu işlem büyük hacimli veri tabanlarında maliyetli bir hal alabilmektedir. Huddar ve Ramannavar (2013:85), 2000 yılında 800,000 petabyte verinin depolandığını iddaa etmektedir.

2015 sıralamalarına bakıldığında en popüler NoSQL veritabanları MongoDB, Apache Cassandra ve Redis olduğu ortaya çıkmaktadır.

2.2. NoSQL Özellikleri ve Avantajları, İlişkisel Veritabanları İle Karşılaştırması

NoSQL her yazılımın olduğu gibi bazı ihtiyaçlar sonucu ortaya çıkmıştır. Bu ihtiyaçlar, uygulamaları kullanıcılar açısından daha çekici kılan ihtiyaçlar olduğu gibi kullanıcı açısından hiçbir anlam ifade etmeyen ama teknik boyutta çok maliyetli ihtiyaçlar olabilir. NoSQL veritabanları, ideoloji olarak ilişkili veri tabanlarının sağlayamadığı bir takım esneklikleri sağlamak üzere geliştirilmiştir.

İlk ve en önemli özelliği genellikle açık kaynaklı olmalarıdır.

NoSQL veritabanları ilişkisel veritabanlarının aksine şemasız veri saklayabilme olanağı sağlar. Bir BİLGİ tablosunu ilişkisel bir veri tabanında ele alınırsa;

Tablo 2
Örnek NoSQL Tablosu

AD	SOYAD	CİNSİYET
Ayşe	Hayriye	Bayan
Ali Veli	Ahmet	Erkek

Şeklinde veriler kaydedilecektir. Bunun anlamı bu tabloda AD, SOYAD, CİNSİYET bilgilerinden farklı bir kişi bilgisine yer verilemeyecek olmasıdır. Bu işlemi gerçekleştirebilmek için BİLGİ tablosunda ALTER işlemi uygulamak gerekecektir ve zamanla bu işlem, eğer çok sık kullanılan riskli bir tablo ise, çok maliyetli olacaktır.

NoSQL veritabanları bu işlemler için esneklik sağlamaktadır. Bu tarz bir işlemde ilk kayıt;

Tablo 3

Örnek NoSQL Tablosu - 2

Ayşe	Hayriye	Bayan
------	---------	-------

şeklinde olacak 2.kayıt ise ;

Tablo 4

Örnek NoSQL Tablosu-3

Ali Veli	Ahmet	Erkek	1.80cm
----------	-------	-------	--------

Olacaktır, yani veri yatay olarak büyümeye açıktır. Herhangi bir ALTER işlemi yapmadan kayıtlarda esneklik sağlanmaktadır. Verinin günbegün artması ve büyümesiyle özellikle sosyal ağ ve bulut işlemlerinde klasik veritabanları zorlanmaya başlanmıştır.

İlişkisel veritabanları ACID prensibi ile çalışmaktadır. Uzunbayır (2015:30), bu prensibin önemli olduğunu, bu şekilde veritabanının işlemlerinin güvenli, güvenilir ve isabetli hale geldiğini vurgulamıştır. ACID özellikleri kısaca;

- Atomicity (Bölünmezlik): Veritabanı gerçekleşen bir işlem ya başarılı olacaktır ya da başarısız olacaktır.
- Consistency (Tutarlılık): Kurallar ve kısıtlamalar veritabanı işlemleri tarafında ihlal edilemez.
- Isolation (İzolasyon): Veritabanı işlemleri birbirinden izole bir şekilde çalışmaktadır.

- Durability (Kalıcılık): İşlemlerin sonuçları kalıcı olmalıdır. Bir güncelleme tamamlandığında sonuçlar sürekli aynı kalmalıdır.

NoSQL veritabanları ise BASE prensibine göre çalışmaktadır. BASE özellikleri kısaca;

- Basically Available (Erişilirlik): Sistem herhangi bir istek olduğunda, istek sonucu başarılı ya da başarısız cevap vermelidir.
- Soft-State (Yumuşak Durum): Veri disk üzerinde kalıcı değildir ama herhangi bir kötü durumda verinin kurtatılması mümkün olmalıdır.
- Eventual Consistency (Dolaylı Kalıcılık): Zamanla gelen veri kalmadığında sistem kalıcı hale geçer.

NoSQL'lerin bir başka özelliği ise kolayca kopyalanabilmeleri ve çok büyük ölçülere ölçeklenebilmeleridir. Büyük veriyi depolayabilmeleri NoSQL'lerin çıkış amaçlarından birisidir. Özellikle büyük resim ve müzik dosyaları ilişkisel veritabanlarında sorunlara yol açabilmektedir. İlişkisel veritabanları bu dosyaları BLOB denilen yapılarda saklamakta ve veri yapısı hakkında herhangi bir bilgiye sahip olamamaktadır.

İlişkisel veritabanları ölçeklendirme sorunları getirebilmektedir. Bu sorunlar donanım desteği almaya yol açar dolayısıyla maliyet getirir.

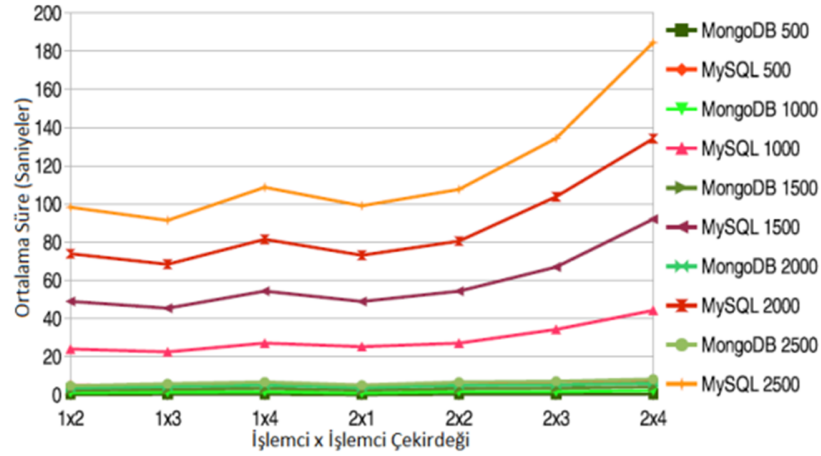
SQL – iyi çalışmakta fakat ölçeklemede sorunlara yol açmaktadır.

NoSQL – iyi çalışmakta fakat bütün iş süreçleri için çözüm olamamaktadır.

NoSQL özelliklerinden bir diğeri de dağıtık çalışabilmeleridir. Dağıtık çalışırken veri bütünlüğünü sağlamaktadırlar. İlişkisel veritabanları bu işlemleri tam olarak sağlayamamakta, sağlayan çeşitleri içinse ciddi iş yükü ve donanım maliyetleri ortaya çıkmaktadır. Yousaf Muhammad (2011), yaptığı incelemede NoSQL'in şema bağımsız bir model sunduğunu ve NoSQL veritabanlarında SQL'e göre verilerin daha hızlı okunup yazıldığını belirtmiştir.

Öztürk ve Atmaca (2016: 204:208), NoSQL veritabanları ile ilişkisel veritabanlarını performans açısından karşılaştırmıştır ve performanslarını incelemiştir.

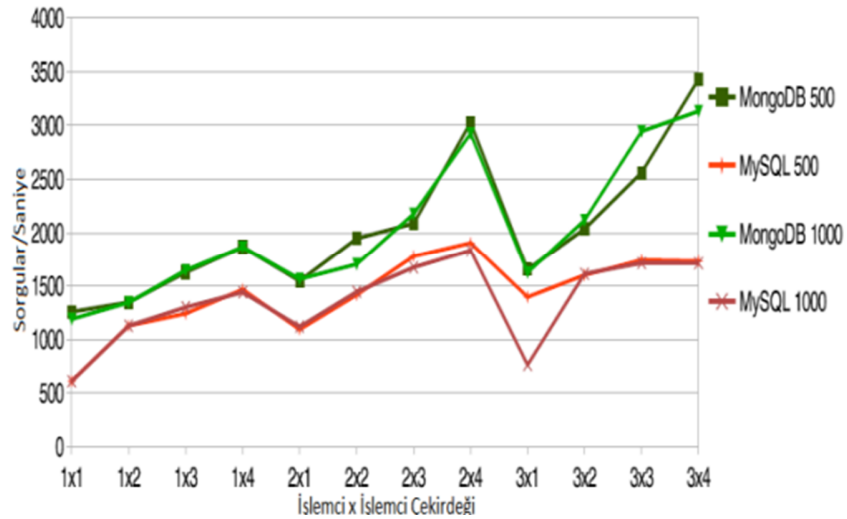
Şekil 19’de MongoDB ve MySQL veritabanlarının detaylı ve karmaşık sorgu kodu ile ortalama süre analizi gösterilmiştir.



Şekil 19: MongoDB ve MySQL Detaylı ve Karmaşık Sorgu Süre Analizi

Kaynak: (Öztürk ve Atmaca, 2016:207)

Şekil 20’da ise MongoDB ve MySQL INNER JOIN ile 500 ve 1000 veri için sorgu/saniye analizi işlemini göstermişlerdir.



Şekil 20: MongoDB ve MySQL Inner Join Karşılaştırması

Kaynak: (Öztürk ve Atmaca, 2016:206)

Yapıları bilinmeyen verilerin saklanması için NoSQL veritabanları çok iyi bir ortam sağlamaktadırlar. Yinede birçok uygulama ve araştırma göstermektedir ki verilen bir kısmını NoSQL veritabanlarında, diğer kısımlarını da data düzenli olarak ilişkisel veritabanlarında saklamak, uygulama açısından en iyi çözüm gibi gözükmektedir. NoSQL veritabanlarında saklanacak veriler, uygulamanın kritik ve uygulamanın hızına etki eden ve ya çok büyük veriler olabilir.

2.3. NoSQL Veritabanı Çeşitleri

NoSQL veritabanları veri saklama ve veri okuma yöntemleri bakımında çeşitlilik göstermektedir. Codemesh 2010'da Ben Scofield NoSQL'lerin kategorize edilmesi hakkında jenerik bir sunum yapmıştır. Popescu, Scofield'un fikrini şöyle sunmaktadır;

Tablo 5
NoSQL Veritabanı Çeşitleri

	Performans	Öçeklenebilirlik	Esneklik	Karışıklık	İşlevsellik
Anahtar-Değer	Yüksek	Yüksek	Yüksek	-	Değişken
Sütun Bazlı	Yüksek	Yüksek	Orta	Düşük	Minimal
Doküman Bazlı	Yüksek	Değişken(yüksek)	Yüksek	Düşük	Değişken(düşük)
Graf Bazlı	Değişken	Değişken	Yüksek	Yüksek	Graf Teorisi

Kaynak : (Popescu,2010)

2.3.1. Anahtar-Değer Tipi Veritabanları

Bu veritabanlarında temel fikir bir hash tablo kullanılmasıdır. Bu Hash Tabloda bir özgün anahtar ile ilişkilendirilmiş veri saklanır. Anahtar-Değer veritabanları en basit ve kolay veritabanlarıdır. Örnek olarak Redis, Memcached, Hazelcast, Amazon SimpleDB gösterilebilir.

2.3.2. Sütun Bazlı Veritabanları

Bu tarz veritabanları çok büyük verileri farklı farklı bilgisayarlarda dağıtık olarak saklayabilmek için oluşturulmuş veritabanlarıdır. Yine bu veritabanlarında da anahtar olmakta ve birden fazla sütünü işaret etmektedirler. Bu veritabanların örnek olarak Cassandra ve HBase gösterilebilir.

2.3.3. Döküman Bazlı Veritabanları

Bu tip veritabanları Anahtar-Değer veritabanlarına benzemektedir. Model temel anlamda versiyonlanmış dökümanlardan oluşmaktadır. Bu dökümanları işaret eden Anahtarlar bulunmaktadır. Bu veritabanları ileri nesil Anahtar-Değer veritabanları olarak görülebilir. Örnek olarak CouchDB ve MongoDB en ünlü veritabanlarıdır.

2.3.4. Graf Veritabanları

Sütun ve satır kullanım yerine daha esnek olan graf modelini temel alan bu veritabanları aynı zamanda birden çok bilgisayara ölçeklenebilmektedir. Önemli örnekler Neo4J, InfoGrid, Infinite Graph

2.4. NoSQL Veritabanları Kullanım Alanları

NoSQL veritabanları günümüzde birçok alanda kullanılmaktadır. Ortaya çıkan iş ihtiyacına göre kullanılan veritabanı tipi değişmektedir. Günümüzde bu kullanılan alanlar sosyal medyadan mobil uygulamalara, e-ticaret uygulamalarından devlet uygulamalarına kadar çeşitlilik göstermektedir. Tablo 6'da veritabanları türleri belirtilmiştir.

Tablo 6
Veritabanı türleri

<u>VERİTABANI</u>	<u>KULLANIM ALANLARI</u>
	Yüksek ölçekte uygulama oturum yönetimleri
	Kullanıcı tercih ve profil bilgileri depolama
	E-ticaret sistemlerinde ürün önerme son gezilen ürünleri kaydetme

Anahtar-Değer veritabanları	ve önerilede bulunma
	Kullanıcı alışveriş tercihlerine göre reklam önermeleri
	Uygulamalar için önbellekleme
Döküman Bazlı veritabanları	Sosyal medya uygulamalarında kullanıcı tarafından üretilen içeriklerin depolamaları
	Uygulamaların loglama işlemleri
	Kullanıcıların alışkanlıklarını takip edebilmek için log işlemleri
	Cihazlardan gelen sensör verileri
Sütun Bazlı Veritabanları	Kullanıcılar arasında mesajlaşma sistemlerinde
	Anlık chat ve mesajlaşma uygulamalarında
	E-Posta işlemlerinde
	SMS işlemlerinde
Graf Veritabanları	En önemli kullanım alanı olarak sosyal medya uygulamaları
	Rota, en yakın yolu bulma gibi işlemlerde dolayısıyla lojistik alanında
	Dolandırıcılık tespit işlemlerinde
	Yatırım yönetimi ve finans alanı
	Ürün önermelerinde

Veritabanları türlerine göre dezavantajlar Tablo 7’de gösterilmiştir.

Tablo 7
Veritabanı türleri ve dezavantajları

<u>VERİTABANI</u>	<u>DEZAVANTAJLARI</u>
Anahtar-Değer veritabanları	Index’leme eksiklikleri
	Veri taramasının kapasitesinde eksiklikler
	Veri üzerinde sorgularda ve CRUD (Create, Read, Update, Delete) işlemlerinde zorluklar
	Veri okuma ve güncellemede verinin yer aldığı dökümanın tamamının okuması ve performans sıkıntıları
	Veri tutarlılığı problemleri

Döküman Bazlı Veritabanları	Join kavramları olmaması
	Bilgisayar belleği kullanımı problemleri
Sütun Bazlı Veritabanları	Verinin yazılması ve güncellemesi maliyeti ve performans problemleri
	Veri sıkıştırma belli veri tiplerinde iyi çalışması
Graf Veritabanları	Bir çoğunun bildirimsel özellikli dili olmaması ve veri sorgusu optimizasyon problemleri
	API kullanım zorunluluğu
	Bir çoğunun değişik platform için yerel implementasyon eksiklikleri olma

2.5. Başlıca NoSQL Veritabanları

2.5.1. MongoDB

MongoDB bütün platformlarda çalışan(cross-platform) bir doküman temelli NoSQL veritabanıdır. İlk olarak 10gen şirketi tarafından 2007 de geliştirilmeye başlanmıştır. Kendi deyimleriyle BSON modelinde bir doküman yapısında çalışmaktadır. MongoDB aynı zaman açık kaynaklı bir yazılımdır ve GNU ve Apache Lisanslarıyla dağıtımı yapılabilmektedir.

BSON yapısı gömülü nesne ve dizileri JSON gibi desteklemektedir.

Herbir dökümanın ID alanı vardır ve bu alan birincil anahtar alanı olarak kullanılmaktadır. Kayıt aramalarını hızlandırmak için yazılım geliştiriciler INDEX denilen anahtarlamalar yapabilmektedir.

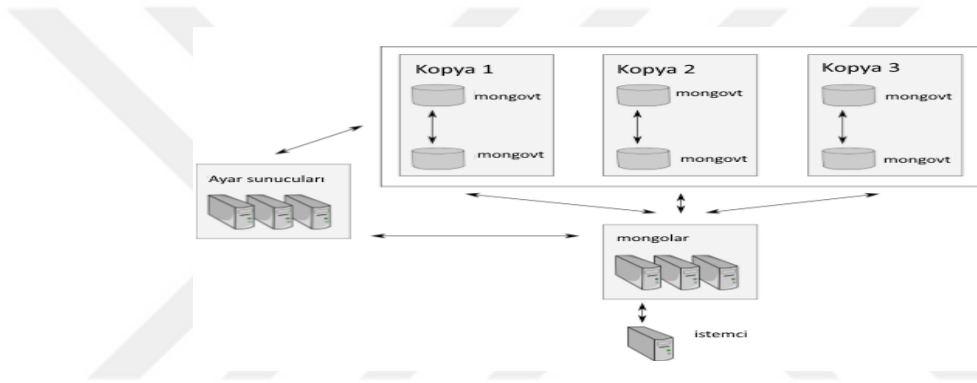
MongoDB sorgulamaları JSON varı bir sözdiziminde BSON nesnesi olarak veritabanı sunucusuna gönderilmektedir. Kullanılan programlama diline göre sorgulama formülasyonları değişkenlik gösterebilmektedir. Mongo sorguları karşılaştırma, şartlı sorgulamalar, mantıksal işlemler, sıralama ve gruptama işlemlerini desteklemektedir.

MongoDB ek olarak daha karışık sorgulamaları MapReduce denilen işlemler ile yapılmasını sağlar. Bu işlem çok büyük veri arasından aranan verileri içeren veri

doküman kümelerini bulma ve daha sonra daha kesin sonuç için bu kümeler arasından işe yaramayan verilere eleme yoluyla istenilen sonuçları bulma işlemidir.

MongoDB aynı zamanda replication yani kopyalama işleminide sağlamaktadır. Bu işlem Master-Slave yani Usta-Köle ilişkisi içeren yapıda olmaktadır (Suter, 2012:5).

MongoDB Sharding denilen kırma yoluyla birden fazla sunucu üzerinde paralel olarak çalışabilmektedir. Bu performans açısından gerçekten önemli bir özellik olmakla beraber birçok ilişkisel veritabanında bu işlem ya olmamakta ya da çok maliyetli olmaktadır. Mongo yapısı klasik olarak aşağıdaki gibidir;



Şekil 21: Mongo Yapısı

Kaynak: (MongoDB, 2016)

2.5.2. CouchDB

Dünyada kullanılan bir diğer popüler veritabanı CouchDB'dir. CouchDB şema bağımsız ve doküman tabanlı bir veritabanıdır ve Apache Kuruluşunun bir parçasıdır. Tamamem Erlang programlama dili ile yazılmıştır aynı zamanda açık kaynak kodludur. Doküman bazlı olan CouchDB, veri saklamak için JSON formatını kullanmaktadır. İlk versiyonu 2005 yılında sunulmuştur.

CouchDB sadece bir NoSQL veritabanı değildir. CouchDB bir web sunucusu olarak kullanılabilir. Bunun en önemli avantajı uygulamalar direk olarak CouchDB'ye entegre olabilmeleridir.

CouchDB web sunucu isteklerinde RESTApi modelinde cevap vermektedir. Dolayısıyla HTTP methodlarına (POST, GET, PUT, DELETE) cevap vermektedir (Tutorials Point, 2015:2).

CouchDB’de yer alan ve verilerin depolandığı dökümanların her birinin özgün numarası vardır. Bunun sebebinin CouchDB’nin replikasyon yönetimi yapmasıdır. Her bir döküman ek olarak revizyon numarası almaktadır. Bu revizyon numarası döküman güncellendiğinde tekrar güncellenmektedir.

CouchDB’de Map-Reduce işlemlerini desteklemektedir. Çok büyük veriler arasından arama ve bu verileri eleme yöntemiyle azaltarak istenilen sonucu bulma işlemini desteklemektedir. Arama süresi veri boyutuyla doğru orantılıdır.

CouchDB “iyimser kopyalama” adı verilen bir kopyalama özelliği kullanmaktadır. Bu kopyalama methodu ile master-master çeşidi kopyalama yapılandırılması kurulabilmektedir. Her bir CouchDB veritabanı diğer bir çalışan örneği ile senkronize edilebilmektedir. Bu özellik manuel olarak tekiklenebilmekte ya da otomatik çalışabilmektedir.

CouchDB içerisinde dâhili olarak bir sharding mekanizması yer almamaktadır. Bunun üzerinde çalışmalar devam etmektedir.

2.5.3. Cassandra

Apache Cassandra açık kaynaklı bir dağıtık veritabanı sistemidir. Çok büyük veriler ile çalışabilmek için geliştirilmiştir. Apache lisanslaması altında olan Cassandra’nın ilk versiyonu 2008 yılında çıkmıştır ve Java ile geliştirilmiştir. Ortaya ilk çıkışı Facebook’ta olmuştur. Facebook arama sonuçlarını hızlandırmak ve iyileştirmek için Cassandra’yı geliştirmeye başlamıştır.

Cassandra sütun bazlı bir veritabanıdır. Cassandra’nın bu özelliği dağıtık kümelerde çalışabilmesini sağlamıştır. Kendine özgü “ring-halka” denilen çalışma mimarisi vardır. Bu mimaride Cassandra’da birbirine bağlı düğümlerden oluşur ve master veritabanı

yoktur. Bu mimarisi ile çok büyük verileri ele alabilmektedir. Bu düğümler arası dağıtık ve ölçeklenebilir bir protokol olan “gossip” ile haberleşmektedir (Hewitt, 2010:15).

Cassandra sharding mekanizması için *partitioner* adında dahili bir bileşene sahiptir. Bu bileşen verinin nasıl dağıtılacağı ve veritabanı kümesinin nasıl olacağını belirlemektedir. Bu bileşen aynı zamanda yeni bir düğüm eklenmesi veya düğüm çıkarılması durumunda veri dağıtımını küme içerisinde düzenlemektedir.

Cassandra’ya erişim arayüzü olarak en çok tercih edilen CQL’dir. Birçok programlama dili için adaptörü bulunmaktadır. Rest API yardımıyla veri okuma ve yazma işlemleri gerçekleştirilebilir.

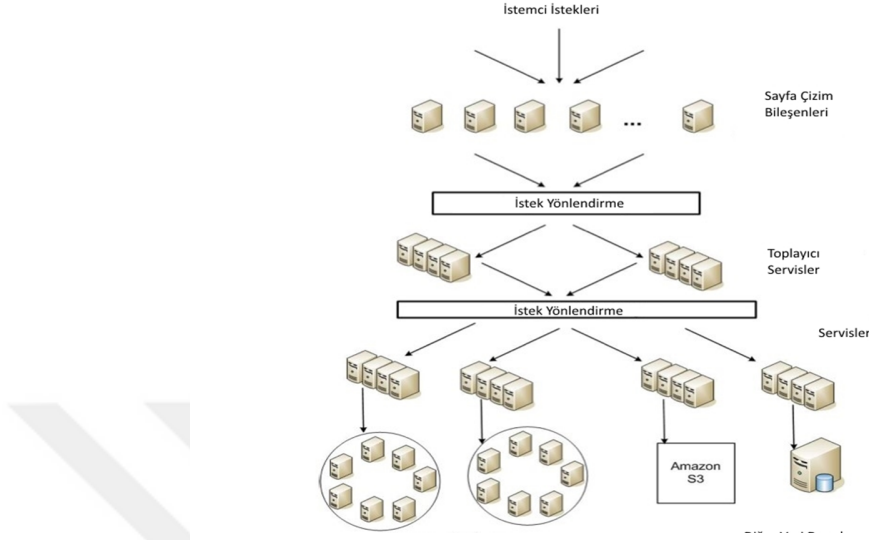
2.5.4. Amazon DynamoDB

DynamoDB, Amazon Web Servisleri hizmeti içerisinde sunulan, anahtar-değer tabanlı dağıtık bir NoSQL veritabanıdır. DynamoDB çok yüksek hız ve erişilebilirlik sağlamakta, kesintisiz şekilde ölçeklenebilmektedir. Basit bir API ile veri okuma ve yazma işlemleri gerçekleştirilebilmekte, tek seferde bir megabte maksimum olacak şekilde veri yükleme yapmaya izin vermektedir.

Dynamo ekosistemi düğümlerden oluşan bir ağ sisteminde çalışmaktadır. Her bir düğümün her an çalışmama ihtimalini göz önünde bulundurmaktadır. Uygulama yüküne göre ek düğümler sisteme eklenebilmekte ve veritabanı kümesi artırılabilir. DynamoDB Amazon bulut sisteminde yer almakta ve kullandığın kadar öde, ihtiyacın olduğunda kullan veya azalt prensibini desteklemektedir. Bu bağlamda teknik anlamda bütün yönetim Amazon tarafından yapılır. Bulut üzerinde çalıştığından erişilebilirlik oranı çok yüksektir.

DynamoDB sunmuş olduğu API ile okuma ve yazma işlemlerini sağlamaktadır. API’de *get(key)* ve *put(key, context, value)* komutları ile bu işlemler gerçekleştirilebilmektedir.

DynamoDB sharding desteği vermektedir. Veritabanında bulunan veriler bir *hash* yardımıyla dağıtılmaktadır. Her bir düğüm belli sayılarda sanal alt düğümlere bölünmüştür.



Şekil 22: Amazon Dynamo DB Yapısı

Kaynak: (Amazon AWS: 2016)

Çoklu master tasarımı ile veriler senkron olarak belli veri merkezlerine yedeklenir. Bu işlem felaket yönetimi, hızlı erişim ve veri güvenliği için yapılmaktadır.

2.5.5. SimpleDB

SimpleDB, Amazon web servisleri hizmeti içerisinde sunulan, şemasız ve dağıtık bir veritabanıdır. SimpleDB'deki veriler farklı veri merkezlerine kopyalanmakta ve bu şekilde verilerin güvenliği ve erişim hızı artırılmış olmaktadır. Erlang programlama dili ile geliştirilmiştir ve ilk olarak 2007 yılında duyurulmuştur.

SimpleDB, kullanıcıya verileri domain'ler arasında orijine etmesine izin verir, bu anlamda ilişkisel veritabanlarındaki tablolar arasında ilişkiye benzemektedir. Yani domain kavramı tablo kavramına eş düşmektedir. Bir satır verinin farklı özellikleri farklı domain yapılarında saklanabilmektedir. Bir domain farklı kayıtların farklı özelliklerini saklayabilmektedir. SimpleDB otomatik olarak sharding mekanizması

içermemektedir. Dolayısıyla çok büyük veriler ölçeklenememektedir. Bu işlemler manuel olarak yapılabilir.

2.5.6. RavenDB

RavenDB, .NET geliştirilmiş, açık kaynak kodlu ve doküman tabanlı veritabanıdır. Veriler şema bağımsız JSON formatında kaydedilmektedir ve .NET için geliştirildiğinde .NET framework'ü içerisinde yer alan LINQ ile veriler üzerinde kolaylıkla işlemler yapılabilir. Aynı zamanda içerisinde barındırdığı sunucu ile RESTFull API yardımıyla veriler üzerinde işlemler yapılabilir. Aynı zamanda kopyalama ve sharding özelliklerine sahiptir.

RavenDB'de kaydedilen her bir dokümanın bir özgün numarası bulunmaktadır ve bu dokümanlar JSON veri formatında kaydedilmektedir. Okuma işlemlerinde hızlı performans göstermektedir.

Veri yedekleme desteği bulunmaktadır. Böylece veri kaybının önüne geçmektedir. Aynı zamanda sharding desteği ile yatay bir şekilde veriyi sunuculara aktarmaktadır. Aynı zamanda trigger'lara sahiptir ve veri yazma ve silme tetikleme yapılabilir.

2.5.7. Google BigTable

Bu hizmet Google'ın NoSQL big data hizmetidir. Bigtable, Google'ın sunmuş olduğu birçok servisin ve hizmetin altyapısını oluşturmaktadır (Analytics, Gmail, Google Earth vs). Bigtable 2004 yılında geliştirilmeye başlanmıştır.

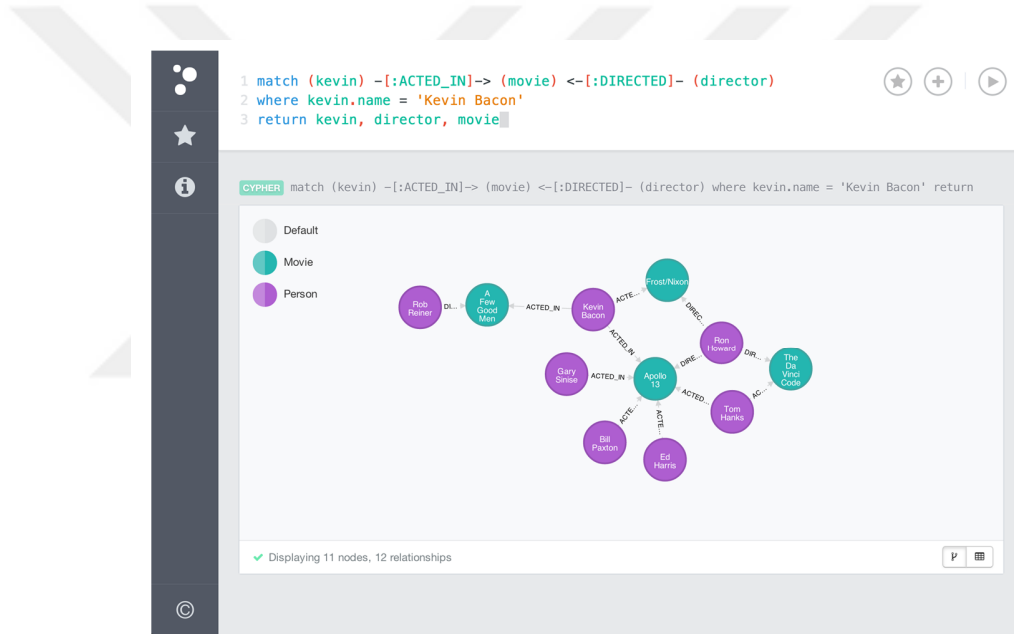
Bigtable kullanıcılara yüksek ölçeklenebilirlik, hız ve performans, en önemlisi yüksek erişilebilirlik sağlamaktadır. Bir bigtable seyrek, dağıtık, kalıcı ve çok boyutludur. İlişki satır anahtarı, sütun anahtarı ve tarih ile indexlenir.

2.6.7. Neo4J

Neo4j, Neo Teknoloji tarafından 2007 yılında geliştirilmeye başlanmış, graf tipinde bir NoSQL veritabanıdır. Java dilinde yazılmıştır. Graf tipinde olan Neo4j aynı zamanda şema bağımsız ve açık kaynak kodludur.

Versiyon 2.1.3, Nisan 2014 ile yaklaşık 35 milyar düğüm, 35 milyar ilişki ve 275 milyar etiket saklama kapasitesine sahiptir.

Neo4j özellikle sosyal medya tabanlı sistemlerde kullanışlı olmaktadır. Bağlı verileri temsil etmek oldukça kolaydır. Aynı şekilde ağaç ve graf yapısındaki verileri okumak çok hızlıdır. Sunmuş olduğu API ile birçok programlama dilleri ile veri iletişimi yapılabilir. Sunmuş olduğu CQL sorgulama dili insanlar için çok kolay okunabilir ve öğrenmesi çok kolaydır, bu açıdan büyük bir avantaj sağlar. Sharding desteği bulunmamaktadır.



Şekil 23: Neo4J Genel Bakış

Kaynak: (Neo4J: 2016)

2.6.8. Elasticsearch

Elasticsearch, Lucene tabanlı bir arama sunucusudur. Dağıtık bir full-text yani tüm-cümle arama motoru olarak çalışır. Bir web arayüzüne sahiptir ve JSON formatında dökümanlara sahiptir. Java ile geliştirilmiş olup açık kaynak kodludur. İlk versiyonu 2010 şubat ayında yayımlanmıştır (Tong ve Gormley, 2015:1).

Elasticsearch herhangi bir dökümanı aramada kullanılabilir. Ölçeklenebilir bir arama motoru sunmaktadır ve dağıtık olarak çalışabilmektedir. Dağıtık olması nedeniyle 0 ya da daha fazla parçacığa bölünebilmektedir ve her bir parçacık 0 ya da daha fazla kopyaya sahip olabilmektedir.

Elastic bulut yardımı ile SaaS olarak kullanılabilmektedir. Günümüzde çok popüler hale gelmiş olup birçok firma tarafından sistemlerde metin aramaları için kullanılmaktadır. Özellikle e-ticaret siteleri, ürün aramalarını hızlandırmak için Elasticsearch'ten faydalanmaktadır.

Elasticsearch sunucu üzerinde çalıştığından uygulamaya erişim sunmuş olduğu API ile JSON formatında veri alışveriş yapılarak sağlanabilmektedir.

BÖLÜM 3: LİTERATÜR TARAMASI

3.1. Bulut Bilişim

Bulut bilişim son 10 yılda popüler olmasına karşın kapsamı oldukça eskidir ve 1960'lara dayanmaktadır. 1961'de John McCarty yapmış olduğu bir konuşmasında ileri bir görüş ortaya koyarak, bilgisayarların ana akım olarak devam etmesi halinde büyük ihtimalle telefon servisi gibi kamuya açık bir hizmet olarak sunulacağını söylemiştir. Böyle olursa, bilişimin endüstriyi çok etkili bir şekilde değiştireceğini iddaa etmiştir (Garfinkel and Abelson, 1999).

J.C.R Licklider (1968), çoklu erişim sistemleri ve kullanıcıyı böyle bir sisteme getireceği yük üzerine belirtilerde bulunmuştur. Hatta bu şekilde bir sistemin maliyetleri hakkında da çıkarımlar yapmıştır.

1990'larda, PC'ler kaynakların paylaşımını ve performansların artırımını sağlayan yerel alan ağları (LAN) üzerinden haberleşmeye başlamıştır. Bunun sonucu olarak kurum ve kuruluşlar kendilerine ait sunucu bilgisayarlardan oluşan bir sistem odası bulundurmaya ve işletmeye başlamışlardır (Bilgi Teknolojileri ve İletişim Kurumu, 2013:5).

Rayport ve Heyward (2009:3), bugünün bilgisayarlar sanki su için duvara musluk takmak ya da prize fişi takarak elektrik elde etmek gibi internet üzerinden akış verdiğini belirtmiştir. Bulutun, hesaplama olanaklarını değiştirerek küçük ve orta ölçekli ticari firmaların güçlü uygulama ve servislere internet aracılığıyla erişim sağlamasına olanak sağladığını belirtmiştir.

Ataş (2013:14), bulut bilişimin ilk olarak nerede kullanıldığına yönelik bir çok tartışmanın olmasına karşın genel bilginin, bulutun ilk olarak telekomünikasyon sektöründe kullanıldığı yönünde olduğundan bahsetmiştir. 1990'ların başında, karmaşık ağ yapılarını içermesi için telekomünikasyon ve ATM altyapılarında kullanıldığını belirtmiştir.

Gong ve diğerleri (2010: 276-278), bulut bilişimin bir takım karakteristik özelliklere sahip olması gerektiğini belirtmişlerdir. Bu özellikler servis tabanlı, gevşek bağlı ve

yüksek tolerans özellikleridir. Ayrıca internet altyapısındaki gibi TCP/IP tabanlı olması gerektiğini vurgulamışlardır.

Perry (2008), bulut bilişim özellikleri olarak kendi kendini iyileştirme, servis düzeyi anlaşması tabanlı, çoklu çalışma (aynı kaynak üzerinde çalışan uygulamaların birbirlerinin verilerine ulaşmadan çalışması), servis tabanlı, sanallaştırılmış, ölçeklenebilir ve veri işleyebilir olması gerektiğinden bahsetmiştir.

Yatırım maliyetini aşağılara çekmek amacıyla kurum ve kuruluşlar bulut bilişim teknolojilerine yönelmektedirler. Ayrıca uydu verileri ya da diğer klasik yöntemler kullanılarak elde edilen çok büyük boyuttaki verilerin özel yazılımlar kullanılarak analiz edilmesine, sonuçların paylaşılmasına ve güvenli ortamlarda saklanmasına gereksinim duyulmaktadır. Bu nedenle bulut teknolojilerini kullanan kurum ve kuruluşlar ile özel firmaların sayısı hızla artmaktadır (Bilgi Teknolojileri ve İletişim Kurumu, 2013:7)

Çatak (2014:5), bulut bilişim kavramının gelişmeye devam ettiğini, sistemin anlamı, kullandığı teknolojiler, kullanım vakaları, riskleri ve faydalarının zamanla daha iyi anlaşılacağını belirtmiştir.

Bichi (2014) yılında yaptığı çalışmada bulut bilişimde etkin kaynak yönetimi üzerine odaklanmıştır. Bu çalışmasında bulut bilişim ortamında tek veya sanallaştırılmış çoklu sunucu içeren sistemlerde kullanım ve doluluk düzeyinin, verimlilik ve zaman gecikmesinin, tepki süresinin nasıl değiştiğine ilişkin matematiksel formüller üzerinde durmuştur. Sanallaştırma teknolojisi bulut bilişim alanında fiziksel sunucu sisteminin faaliyetlerini tamamlamak için kullanıldığını, çalışmasında farklı dengeleme teknikleri kullanan sanal kaynak sistemi için uygun yük dağıtımını amaçlayan “Max-Min” algoritmasına dayalı etkin bir yapı geliştirmeye çalışmış ve sonuçları simülasyonla göstermiştir.

Literatürde yapılan bazı çalışmalar şöyledir;

Chandran ve Angepat (2010), yayınlamış oldukları çalışmada bulut bilişim çevresinde dahil olan riskleri analiz etmişlerdir. Bu çalışma farklı türdeki riskleri özetlemiş ve bu risklerin varlığının bulut kullanıcılarını nasıl etkileyeceğini analiz etmişlerdir

Sevli (2011), yüksek lisans tezinde bulut bilişim kavramlarını detaylı bir şekilde ele almış, Windows Azure üzerinde bulut bilişim temelli bir eğitim portalı geliştirmiştir. Bu portal çalışması ile eğitiim kurumlarının altyapı konusundaki kaygılarını gidermeyi amaçlamıştır.

Ataş (2013) yapmış olduğu yüksek lisans tezinde bulut bilişim platformlarının performans değerlendirmesi üzerinde durmuştur. Yaptığı çalışmada işlem gücü, veritabanı işlemleri ve ana hafıza bant genişliği işlevlerini test etmiş ve sonuçları birbirleriyle kıyaslamıştır.

Görelık (2013) yılındaki çalışmasında bulut bilişim modellerini incelemiştir. Çalışmasında bulut bilişim modelleri ve gelecekteki potansiyel bulut trendleri üzerinde durmuştur. Çalışması bulut bilişim servisleri (SaaS, PaaS, IaaS) ve yerleştirme modellerinin kıyaslamalı analizini içermektedir.

Çatak (2014) yılındaki doktora tezinde bulut bilişim sistemlerinde eşle/indirme yöntemiyle veri madenciliği yazılım çatısı geliştirmiştir. Bu çalışma ile çok sınıflı destek vektör makinesi algoritmasının bulut bilişim sistemleri üzerinde eşle/indirge tekniği ile eğitilmesini incelemiştir.

Khanghahi ve Ravanmehr (2013), çalışmasında bulut bilişim performans değerlendirmesi üzerinde durmuş, sorunlar ve çözümleri için önerilerde bulunmuştur. Bulut bilişimdeki büyük sorunlar için senaryolarak oluşturarak performans çalışması yapmış, bu çalışmas sonucunu analiz etmişlerdir.

Nwobodo (2015), yayınladı çalışmada bulut bilişim platformlarını karşılaştırmıştır. Çalışmada AbiCloud, Eucalyptus, Nimbus ve OpenNebula platformlarını karşılaştırmalı incelemiştir.

Kriushanth, Arockiam ve Mirobi (2013), bulut bilişimde otomatik ölçekleme üzerinde bir çalışma yapmışlardır. Bu çalışmalarında bulut bilişime genel bir bakış yapmış ve otomatik ölçeklendirmenin önemini vurgulamışlardır.

Bediroğlu (2013), yüksek lisans tezinde web haritalarının bulut üzerinde yayınlanması ile ilgili çalışmış ve Trabzon ili örneğini konu edinmiştir. Çalışmasında bulut bilişim ve konumsal bulut bilişim teknolojilerinin; günümüzdeki avantajlarının ve dezavantajlarının incelenmesini amaçlamıştır. Çalışmada konumsal verilerin görüntülenmesi ve coğrafi bilgi sistemi analizlerini, konumsal bulut bilişim teknolojisi ile yapmıştır. Durumu Türkiye açısından incelemiş, yaşanabilecek sorunlar ve çözümleri tespit etmeye çalışmıştır.

3.2. NoSQL

1998 yılında bir ilişkili veritabanı yönetim sistemi, İtalyan Carlo Strozzi tarafından geliştirilmiş ve GNU lisansı altında yayınlanmıştır. Çalışmasını NoSQL olarak adlandırmıştır ve bilerek SQL’de bulunan veri yapısı ve veri değiştirme yapısından kaçınmıştır. Bunun yerine veriyi açık metin dosyaları şeklinde, her bir dosyanın bir tablo olarak değerlendirildiği bir dosya sisteminde saklamıştır (Henricsson, 2011:12-13).

Muhammad (2011:4), klasik ilişkili veritabanlarının büyük ve tüm verileri ölçeklemede çok zorlandığını ve bu işlemin üstesinden gelmenin bir yolunun NoSQL veritabanları olduğunu belirtmiştir.

Näsholm (2012:1), günümüzde modern uygulamaların gereksinimlerinin, özellikle 90’lar ve 2000’lerin başı itibarıyla web kavramının artmasıyla değiştiğini belirtmiştir. Büyük web sitelerinin milyarlarca kez sayfaları hizmet olara sunmak zorunda olduğunu ve kullanıcıların bu sayfalara hızlı bir biçimde erişmek istediğini, bazı gereksinimlerin ilişkili veritabanları ile tatmin edici bir şekilde karşılanamadığını vurgulamıştır.

İlişkisel veritabanı yönetim sistemi, yani tabloları ve ilişkileri kullanarak veri kaydetmek her zaman en iyi çözüm olmayabilir. Normalleştirilmemiş ve ilişkisel yapıya uymayan veriler yeniden yapılandırılmak zorundadır. Bu işlemi gerçekleştirmek için

geliştirilmesi gereken kod ise zaman alacaktır. Verileri tekrar formatlamak aynı zamanda kodun yavaşlamasına sebep olacaktır (Henricsson, 2011:13).

NoSQL veritabanları BASE kavramı (temelde erişilebilir, yumuşak durumda, en sonunda kalıcı) üzerine, yüksek erişilebilirlik ama tutarlılıktan feda eden bir yapı ile çalışmaktadır. Diğer tarafta İlişkisel veritabanları ise ACID (Atomik, Tutarlı, Korunmuş, Dayanıklı), yani bütün işlemlerin doğru tamamlandığı, veritabanı yapısını bozmadığı ve verinin tutarlı olduğu yapı ile kurulmuştur (Abramova, Bernardino ve Furtado, 2014: 18).

Jablonski ve Hecht (2011:366), web üzerindeki verinin artmasının başarılı web siteleri (Facebook, Amazon, Google) bir problem olarak değerlendirildiğini belirtmiştir. Petabyte’larca veri ile başa çıkmak, bu verileri okumak ve isteklere geciktirmeden cevap vermek gerekmektedir. Bu gereksinimlerle başa çıkmak için binlerde donanım kümesi yönetildiğini söylemiştir. Nomalize olmuş veri modelleri ve tümüyle ACID desteği ile bu alanı elverişli hale getirmenin mümkün olmadığını, çünkü tablo birleştirme ve kitlemenin performansa negatif yönde etki ettiğini vurgulamıştır.

Literatürde yapılan bazı çalışmalar şöyledir;

Muhammad (2011), dağıtık NoSQL veritabanları üzerinde bir değerlendirme çalışması yapmış ve bu dağıtık veritabanlarının MMO oyunlarında kullanılması üzerinde durmuştur. Çalışmasında NoSQL ve SQL veritabanları arasındaki en güvenilir olanını bulmaya çalışmıştır.

Aydemir (2016) yüksek lisans tezinde sınır güvenliği için gerçek zamanlı akan veri ve yığın veri üzerinde analiz yapmayı sağlayan tutarlı, dayanıklı, dağıtık ve ölçeklendirilebilir bir sistem ilk örneği geliştirmiştir. Geliştirilen sistemin test edilebilmesi için örnek senaryolar oluşturmuş elde edilen sonuçları gözlemlemiştir.

Brad (2012), yayınladığı çalışmasında NoSQL veritabanlarında veri tekilleştirmesi üzerinde durmuştur. Bu çalışmasında tekrarlayan verileri maksimum şekilde tekilleştirmeyi performansı artırmayı amaçlamıştır.

Roijackers (2012), yüksek lisans tezinde SQL ve NoSQL veritabanları arasında bölüştürülen verilerin getirmiş olduğu zorlukları aşmak amaçlı bir soyut veritabanı geliştirmiştir.

Shermin (2013), yapmış olduğu tezde NoSQL veritabanlarındaki klasik rol tabanlı erişim kontrolünü geliştirerek bağlam bazlı soyut bir model sunmuştur. Sunulan modelin güvenlik politikalarını değerlendirmekte ve çalıştırmakta olduğunu belirtmiştir.

Atmaca ve Öztürk (2016) yayınında ilişkisel ve ilişkisel olmayan veritabanı sistemlerinin mimari performanslarını incelemiştir. Çalışmada veri tabanı modelinin seçilmesi ve ölçütlerinin ortaya koyulmasıyla en uygun performans ölçümleri, veri tabanlarının avantaj ve dezavantajları ile oluşturulacak yapının uygun hale getirilmesi hakkında bilgiler vermişlerdir.

Näsholm (2012), tezinde NoSQL veritabanlarının şema bağımsız ve yatay ölçeklenmesinin bazı uygulamalarda veri yazma ve okumalarda problemlere yol açtığı üzerinde durmuştur.

Hecht ve Jablonski (2011), yayınlarında NoSQL veritabanı seçmenin zorluğuna vurgu yapmış ve bu veritabanlarının uygulanabilirlik, veri modelleri, sorgulama olasılıkları, eşzamanlılık gibi özelliklerini karşılaştırmıştır.

Uzunbayır (2015) tezinde farklı veritabanı yönetim sistemleri ve zorluklarını özetlemiştir. İlişkisel ve grafik tabanlı olmak üzere iki farklı veritabanı teknolojisini analiz etmiş ve karşılaştırmıştır. Bu iki teknoloji için sosyal ağ ile çevrimiçi alışveriş uygulaması olan TrendPin'in üzerinde veri modelleri tasarlamış ve geliştirmiştir.

Orend (2010), nesne ilişkili kalıcı katmanın yerine konulması için NoSQL veritabanlarının analizi ve sınıflandırılması, yeteneklerine göre değerlendirilmesi çalışmasını tezinde konu edinmiştir. Bu çalışmada NoSQL ve benzer ilişkili veritabanlarını karşılaştırmıştır.

BÖLÜM 4: ÖRNEK UYGULAMA – TÜRKİYE’DE 1970-2014 YILLARI ARASINDA YAŞANMIŞ TERÖR OLAYLARININ MICROSOFT AZURE BULUT PLATFORMU ÜZERİNDE NEO4J NOSQL VERİTABANI MODELİ İLE AMPİRİK DEĞERLENDİRİLMESİ

Terör olayları geçmişten günümüze ülkeler ve toplumlar için büyük sorunlar teşkil etmiştir. Birçok ülke zaman zaman terör saldırılarına maruz kalmış ve kalmaya devam etmektedir. Bu olaylar ülkelerin ekonomisi olumsuz etkilemekte ve toplum üzerinde korku salmakta ve huzursuzluk yaratmaktadır. Bazı kaynaklara göre Türkiye, geçmişten günümüze kadar terör olayları ile mücadelede 350 milyar dolar üzerinde para harcamıştır. Bu uygulama ile 1970-2014 yılları arasında Türkiye’de gerçekleşmiş terör olaylarının Graph veri tabanı ile modellemesi yapılmıştır. Bu model yardımı ile verilerden görsel modeller oluşturularak olayların yorumlanması ve ileride gerçekleşebilecek olaylar için önlem alınması amaçlanmıştır. Aynı zamanda Microsoft Azure bulut platform kullanılarak bulut teknolojilerinin avantajlarından faydalanılmıştır.

4.1. Terör Verilerinin Elde Edilmesi

Modelleme 1970 ve 2014 yılları arasındaki Türkiye’de gerçekleşmiş terör olayları verilerine Maryland Üniversitesinin yönettiği Global Terrorism Database üzerinden ulaşmak mümkündür. Yerel olayların yanı sıra, Amerika için, uluslararası veriler de kaydedilmektedir. Yaklaşık olarak 150 bin kayıtlı olay bulunmaktadır. Global Terrorism Database çalışmalarına göre veriler şöyledir;

- 150.000 üzerinde terör olayı
- 75.000 bombalama, 17.000 suikast, 9.000 adam kaçıрма olayı,
- Her bir olay için en az 45 bilgi,
- Veri elde edilmesi için 1998 ve 2015 arasında 4 milyondan fazla kaynak incelenmiştir.

4.2. Elde Edilen Terör Verilerinin Analizi ve Sistem Modellemesi

Üniversitenin ilgili sitesinden verilere CSV formatında ulaşmak mümkündür. İndirilen veriler tüm dünyada gerçekleşmiş olayları barındırır ve arasından Türkiye’de gerçekleşmiş olanlar ayıklanarak yeni bir CSV dosyası oluşturulmuştur.

Elde edilen CSV formatında dosyada yer alan bazı başlıklar Şekil 24’de gösterilmiştir.

1	eventid	year	imonth	iday	approx	extend	resolut	country	country	region	region_txt	provstate	city
2360	2359	1998	2	27		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2361	2360	1998	4	10		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2366	2365	1998	8	29		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2370	2369	1998	9	19		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2375	2374	1998	11	5		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2378	2377	1998	11	13		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Eminonu C
2384	2383	1999	1	12		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Uskudar
2387	2386	1999	3	3		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2388	2387	1999	3	3		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2389	2388	1999	3	3		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2390	2389	1999	3	3		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2391	2390	1999	3	3		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2393	2392	1999	3	4		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2395	2394	1999	3	6		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul
2396	2395	1999	3	6		0		209	Turkey	10	Middle East & North Africa	Istanbul (Province)	Istanbul

Şekil 24: GDT CSV Verileri Bazı Sütun Başlıkları

Mevcut veriler ve başlıkları İngilizce’dir. İngilizce başlıklar Türkçe’ye çevirilerek model oluşturulmuştur. İndirilen veri setinde 127 adet başlık bulunmaktadır. Toplam olarak GTD’de kayıtlı ve Türkiye’de gerçekleşmiş olan 3144 adet terör olayı yer almaktadır. Oluşturulacak model 11 ana başlık içermektedir ve kalan başlıklar bilgi amaçlı kullanılmıştır. Tablo 8’de bazı önemli başlıklar Türkçe olarak açıklanmıştır.

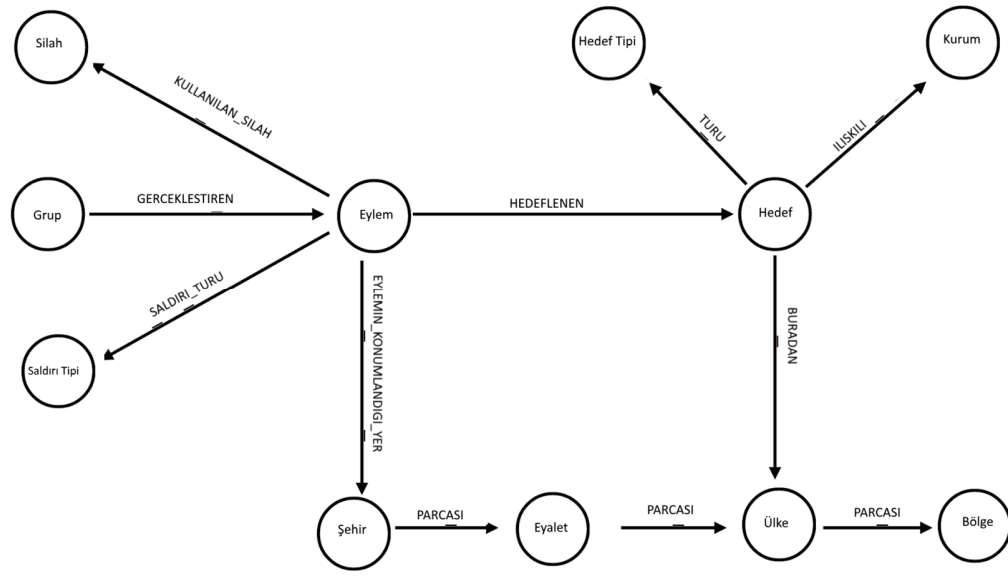
Tablo 8

GDT CSV Verileri Önemli Sütun Başlıkları

BAŞLIK	AÇIKLAMA
ulke	Olayın gerçekleştiği ülke numarası
ulke_metin	Olayın gerçekleştiği ülke ismi
bolge	Olayın gerçekleştiği bölge numarası

bolge_metin	Olayın gerçekleştiği bölge ismi
eyalet	Olayın gerçekleştiği yöre
sehir	Olayın gerçekleştiği şehir
saldiritipi1, saldiritipi2, saldiritipi3	Yapılan saldırıların tip numarası
saldiritipi1_metin, saldiritipi2_metin, saldiritipi3_metin	Yapılan saldırıların metni
hedefitipi1, hedefitipi3, hedefitipi3	Hedeflenen varlıkların numaraları
hedefitipi1_metin, hedefitipi2_metin, hedefitipi3_metin	Hedeflenen varlıkların metni
kurum1, kurum2, kurum3	Hedeflenen kurumların isimleri
uyruk1, uyruk2, uyruk3,	Hedefin uyruk numaraları
grupadi, grupadi2, grupadi3	Saldırıyı gerçekleştiren grup
silahitipi1, silahitipi2, silahitipi3, silahitipi4	Saldırıda kullanılan silahların numaraları
silahitipi1_metin, silahitipi2_metin, silahitipi3_metin, silahitipi4_metin	Saldırıda kullanılan silahların metinleri
eylem	Saldırının kendisidir, eylemid, tarih, koordinat, özet, silahların detayları, yaralı sayısı, ölü sayısı gibi kalan bilgilere sahiptir

Elde edilen veri setindeki verilerden sistemin modeli oluşturulmuştur. 11 ana başlık ile model yapısı Şekil 25 'de gösterilmiştir.



Şekil 25: Model Yapısı

Model merkezinde Eylem yani terör olayının kendisi yer almaktadır. Bir Eylem, tarih, ölü sayısı, fidye vb gibi bir çok bilgiye sahiptir. Eylem, olayda kullanılmış Silah node' u ile KULLANILAN_SILAH etiketi üzerinden, Eylem'i gerçekleştiren Grup ile GERCEKLESTIREN etiketi üzerinden ve SaldırıTipi ile SALDIRI_TURU etiketi üzerinden ilişkilidir. Bir Eylem bir Hedef'i hedef almaktadır HEDEFLENEN etiketi ile Hedef node'una bağlıdır. Hedef ise HedefTipi ve Kurum node'ları ile ilişkilidir. Bu ilişkiler sırasıyla TURU ve ILISKILI etiketleri ile sağlanmıştır. Bir Eylem gerçekleştiği yer olarak EYLEMIN_KONUMLANDIGI_YER etiketi ile Şehir node'una bağlıdır. Şehir Eyalet ile, Eyalet Ülke ile, Ülke ise Bölge node'una PARCASI etiketi ile bağlıdır. Aynı zamanda Hedef BURADAN etiketi ile Ülke node'u ile ilişkilidir.

Elde edilen veriler Neo4j ortamına aktarılıp ilişkilendirilmiştir. Veriler CSV dosyasından okunarak node'lar oluşturulmuş ve bu node'lar birbirlerine bağlanmıştır. Aşağıda ülke ve bölgelerin CSV dosyasından okunarak node'ların oluşturulması ve birbirlerine bağlanması örneği kod ile gösterilmiştir;

```

load csv with headers from "file:/gtdb_transform.csv" as csv
fieldterminator ';'
with distinct toInt(csv.ulke) as ulke, csv.ulke_metin as ulke_metin

```

```
create (c:Ulke {id: ulke, name: ulke_metin });
create index on :Ulke(id);
load csv with headers from "file:/gtldb_transform.csv" as csv fieldterminator ';'
with distinct toInt(csv.bolge) as bolge, csv.bolge_metin as bolge_metin, toInt(csv.ulke) as ulke
match (c:Ulke {id: ulke})
merge (r:Bolge {id: Bolge, name: bolge_metin })
merge (c)-[:PARCASI]->(r);
```

Modelde bir olay bir şehirde gerçekleşmekte, şehir bilindiğinde ilgili ülke de bulunabilmekte, bu olayda kullanılan silah ve olayın tipi bilinmekte, olayda hedef alınan kitle veya kurum bilgisine ulaşılabilir. Aynı zamanda bu olayı gerçekleştiren terör grubuna da olay üzerinden ulaşılabilir. Hedef ile ülke arasındaki ilişki sayesinde hedef alınan kitle veya kurumun hangi ülkeye ait olduğu bulunabilmektedir.

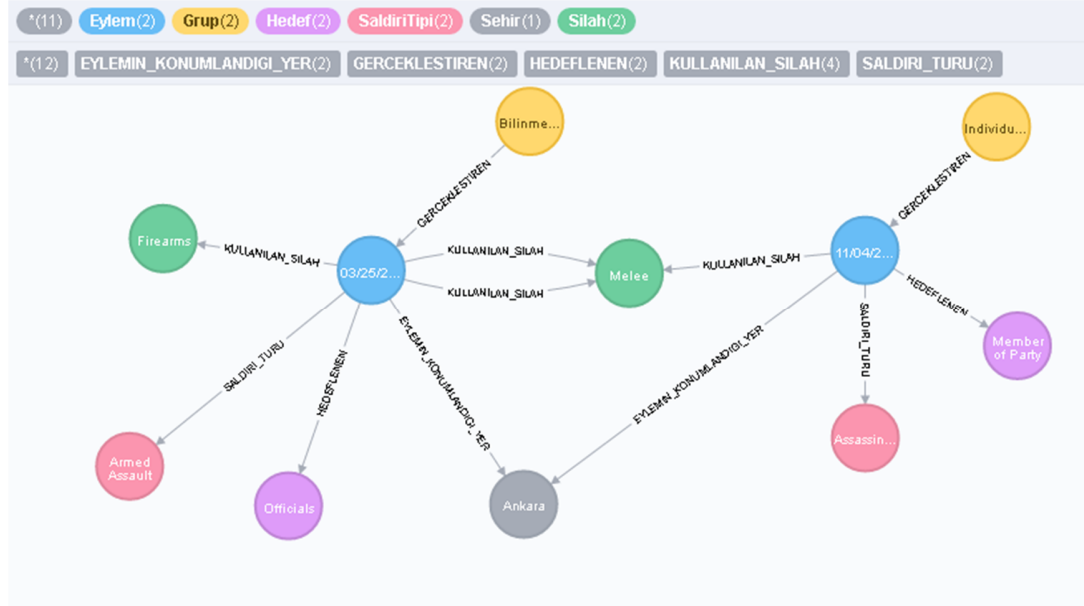
4.2. Model Üzerinde Sorgulamalar

Neo4j 'nin Cypher sorgulama dilini kullanarak hem görsel hem de satır bazlı sorgulamalar yapılabilir. Gerçekleştirilen modellemelerden birçok sorgulama yapmak mümkündür. Birkaç örnek ile bakılırsa;

- Şehir, ülke, bölge, silah, hedef, hedef tipi, grup, saldırı tipi, kurum bazlı sorgulamalar,
- Belli bir tarih aralığında, belli bir şehirde, belli saldırı tipindeki olayların sorgulanması,
- Belli saldırı tiplerinin en çok gerçekleştirildiği yerlerin sorgulanması.

2014 yılında Ankara'da ölüm bilgisi kaydedilmiş olay sorgu sonucu Şekil 26'de gösterilmiştir.

```
match (e:Eylem)-->(c:Sehir) where e.year=2014 and e.kill is not null and c.name="Ankara" return e order by e.kill desc
```

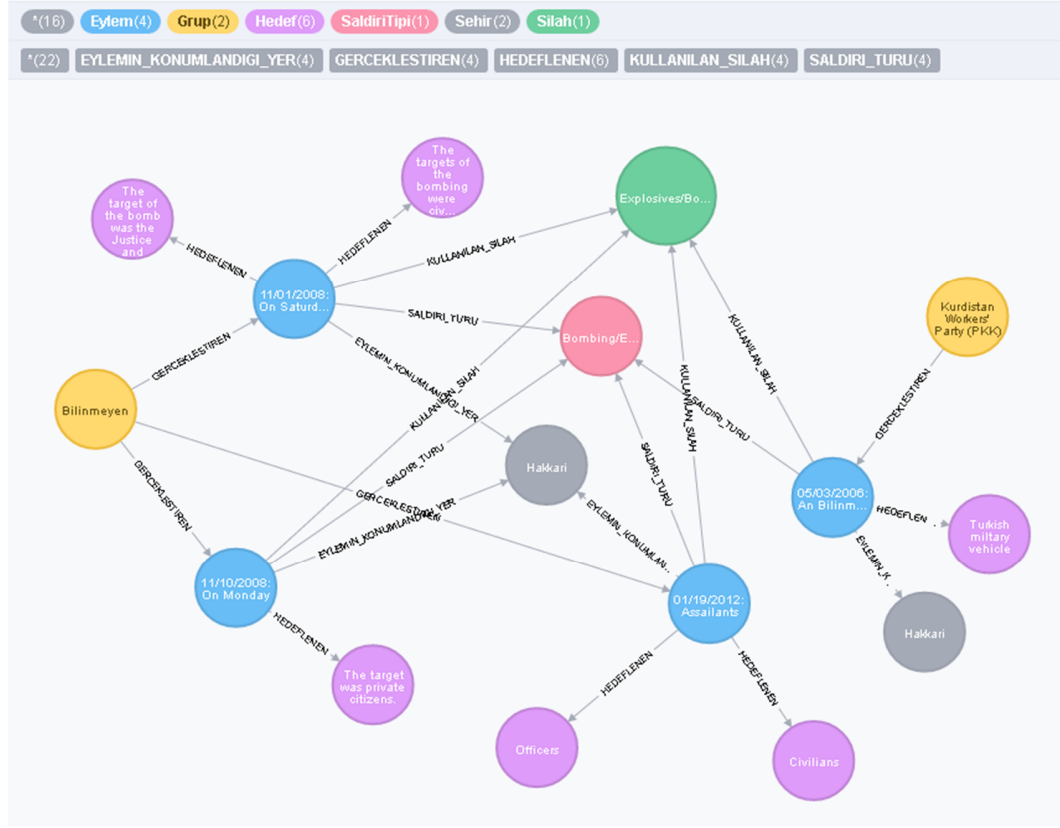


Şekil 26: 2014 Yılında Ankara’da Gerçekleşmiş Ölümlü Terör Eylemleri

2014 yılındaki bu 2 olayın saldırı tipi silahlı saldırı ve suikast olduğu kırmızı ile görülmektedir ve her iki saldırıda da yakın dövüş silahı kullanılmıştır. Suikast olarak gerçekleştirilen olayın belirsiz kişilerce yapıldığı görülmekte, silahlı saldırı olarak gerçekleştirilen olayın ise kim tarafından gerçekleştirildiği bilinmemektedir.

2005 ve 2014 yılları arasında Hakkâri’de Bombalama/Patlama tipinde gerçekleştirilen terör olaylarının sorgu sonucu Şekil 27 ile gösterilmiştir.

match (at:SaldırıTipi)--(e:Eylem)-->(c:Şehir) where e.year<2014 and e.year>2005 and at.id=3 and c.name="Hakkari" return e



Şekil 27: 2005 ve 2014 Yılları Arasında Hakkari’de Gerçekleşen Bombalama/Batlama Tipindeki Saldırıları

Sorgu sonucunda 4 adet eylem olduğu görülmektedir ve bunlar mavidir. Saldırıların 2 farklı grup ile parti binaları, siviller, askeri araçlar ile memurlara yönelik olduğu görülmektedir.

1974 ve 2014 yılları arasında Bombalama/Patlama tipinde saldırıların en çok görüldüğü ilk 10 şehir Şekil 28 ile gösterilmiştir.

match (at:SaldiriTipi {id:3})<-(e:Eylem)-->(c:Şehir) where e.year>1974 and e.year<2014 return c.name AS 'Şehir Adı',at.name AS 'Saldırı Şekli',count(e) AS 'Saldırı Sayısı' order by count(e) DESC

Şehir Adı	Saldırı Şekli	Saldırı Sayısı
İstanbul	Bombing/Explosion	545
Ankara	Bombing/Explosion	133
İzmir	Bombing/Explosion	58
Bilinmeyen	Bombing/Explosion	53
Adana	Bombing/Explosion	38
Diyarbakır	Bombing/Explosion	24
Antalya	Bombing/Explosion	15
Semdinli	Bombing/Explosion	15
Cizre	Bombing/Explosion	10
Batman	Bombing/Explosion	9

Şekil 28: 1974 ve 2014 Yılları Arasında Bombalama/Patlama Tipinde Saldırıların En Çok Görüldüğü İlk 10 Şehir

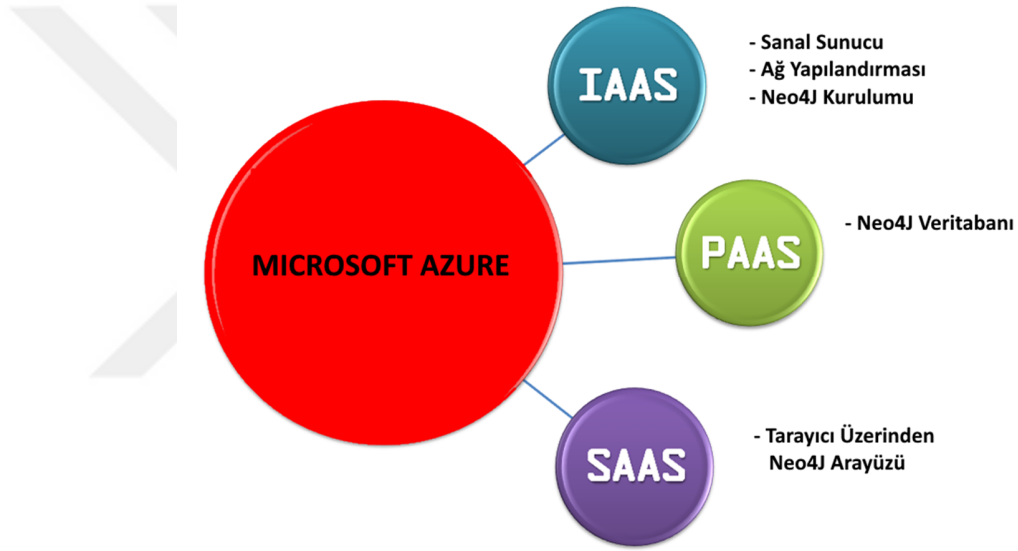
Sorguya göre en çok patlamalı saldırı 545 ile İstanbul olduğu görülmektedir. Bu sıralamayı Ankara, Şemdinli, Diyarbakır ve Yüksekova izlemektedir. Diğer şehirler ile birlikte toplam saldırı sayısı 121'dir.

4.3. Microsoft Azure Bulut Platformunda NoSQL Çalıştırılması

Bulut platformu kullanılmasındaki ana sebep bulut teknolojilerinin getirmiş olduğu temel avantajlardan yararlanmaktır. Oluşturulan model 3 binden fazla kayıt bulunmaktadır ve bunların birbirleri ile ilişki sayısı oldukça yüksektir. Bu modelde arama yapmak, ilişkileri sorgulamak, yapılan sorgulamaya göre uzun sürebilir. Sistemdeki kayıt ve kullanıcı sayısı arttıkça bu sorgulama süresi de artacaktır. Bu sorunlar göz önünde bulundurulduğunda bulut teknolojilerinin getirmiş olduğu ölçekleme özelliğinden yararlanılabilir. Kullanılan işlemci gücü ve sayıları, depolama alanları, internet bant genişliği artırılmak suretiyle performans artırılabilir. Aynı zamanda bulut bilişim kullanıma bağla ücretlendirme yapacağından, işlemci kullanım miktarı ve sorgulamanın kullandığı bant genişliği ne kadar olursa, o kadar ücret ödenecektir. Bu anlamda belli sayıda sunucu kiralayıp hiç çalıştırmadan ücret

ödenmesinin önüne geçilebilmektedir. Bu işlemler yapılırken sunucu bakımlarıyla uğraşmadan sistem kullanılmaya devam edilebilir.

Microsoft Azure bulut bilişim platformu, birçok NoSQL veritabanını desteklemekte ve bunlardan birisi de Neo4J'dir. Azure, IaaS ve SaaS hizmetinde bulut seçenekleri sunduğundan, istenirse sanal sunucu altyapı olarak kiralanıp Neo4J manuel olarak kurulabilmektedir. Sistemin yapılandırılması için bulut bilişimin 3 hizmet modelinden de faydalanılmıştır. Sistem yapısı aşağıdaki şekilde gösterilebilir;



Şekil 29: Azure Hizmet Modelleri ve Oluşturulan Sistem

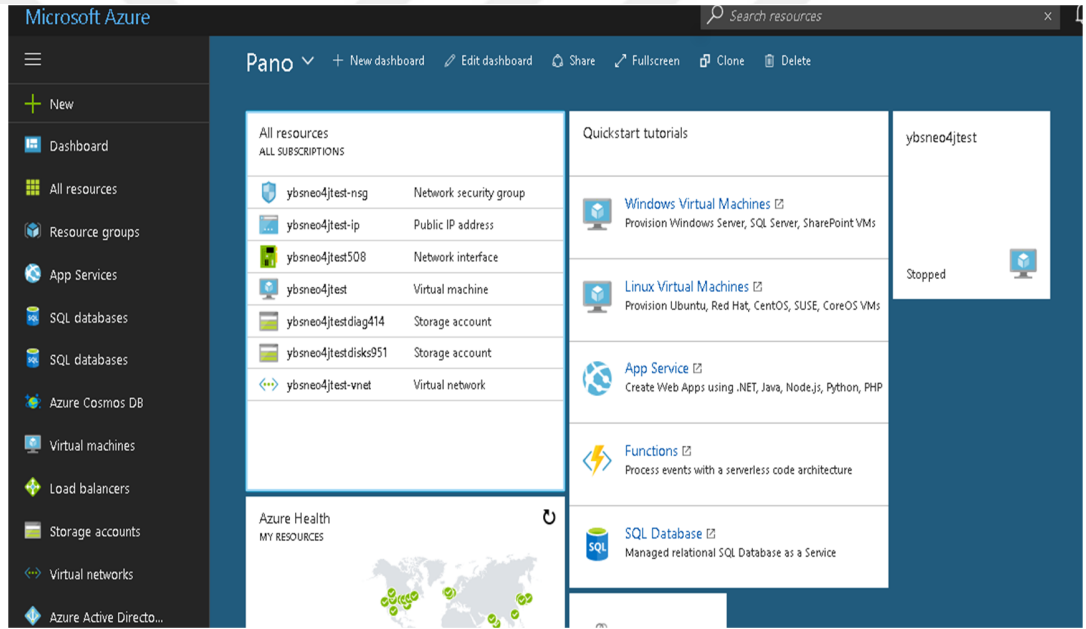
Sistemin bulut bilişim hizmet modellerine göre oluşturulması şöyledir;

- **IaaS hizmet modeli:** Azure portalı üzerinden Windows Server 2012 işletim sistemli bir sanal sunucu oluşturulmuştur. Sanal sunucu lokasyon olarak batı avrupada oluşturulmuştur. Sunucuya uzaktan bağlanılarak Neo4J veritabanı <https://neo4j.com/download/community-edition/> bağlantı adresi üzerinden indirilerek kurulumu yapılmıştır. Bu kısım IaaS kısmında yer almaktadır.
- **PaaS hizmet modeli:** Azure portalı üzerinden veritabanına erişim için gerekli ağ yapılandırılmaları yapılmıştır. Neo4J varsayılan olarak 7474 portu üzerinde

çalışmaktadır. Neo4J için de gerekli ağ yapılandırmaları yapılarak sunucu üzerinde uzaktan erişilebilir hale getirilmiştir. Yapılandırma için kullanılan kaynak web siteleri kaynakça’da gösterilmiştir. Böylece Neo4J PaaS modelinde hizmet verir hale getirilmiştir.

- **SaaS hizmet modeli:** Neo4J bu yapılandırma ile browser üzerinden erişilebilir hale gelmiştir. Sistem isim yapılandırmasıyla <http://ybsneo4j.westeurope.cloudapp.azure.com:7474/browser/> adresi üzerinde hizmet verir hale getirilmiştir. Bu sayede SaaS modeli hizmete sunulmuştur.

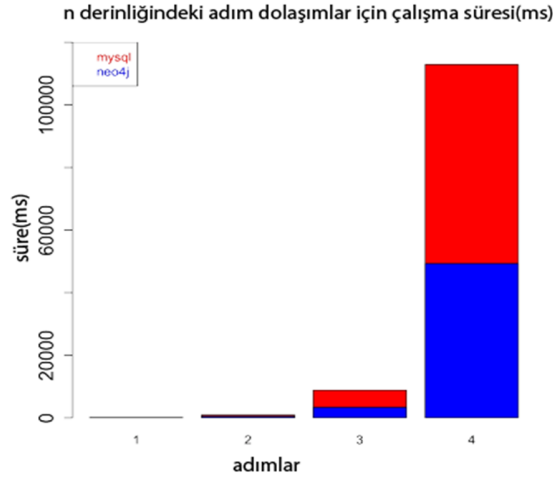
Azure portal, yapılandırmalarından sonra aşağıdaki şekilde görülmektedir;



Şekil 30: Azure Portal Yapılandırılmış Hali

4.4. Performans Analizi

Neo4j 3.x versiyonu ile birlikte sınırsız node, ilişki, özellik desteklemektedir. Bu rakam oldukça yüksek bir rakamdır. Rodriguez (2011) de Neo4j ve MySQL veritabanlarını karşılaştırmış ve performans analizleri yapmıştır. Bu Neo4j ve MySQL kayıt okuma hızı karşılaştırması Şekil 30’da gösterilmiştir.



Şekil 31: Neo4j ve MySQL Performans Karşılaştırması

Kaynak: (DZone: 2011)

Rodriguez'in MySQL ve Neo4j kayıt sayılarının artışında harcanan sürelerdeki tespiti Tablo 9'da gösterilmiştir.

Tablo 9

Neo4j ve MySQL Performans Karşılaştırması

	MYSQL	NEO4J
Adım 1 – süre/kayıt	124ms/11360	27ms/11360
Adım 2 – süre/kayıt	922ms/162640	474ms/162640
Adım 3 – süre/kayıt	8851ms/2206437	3366ms/2206437
Adım 4 – süre/kayıt	112930ms/28125623	49312ms/28125623
Adım 5 – süre/kayıt	N/A	862399ms/358765631

Kaynak: (DZone: 2011)

Performans rakamları donanıma bağlı olduğundan tam rakam vermek mümkün değildir. Neo4j milyar düzeyinde node tutabildiği gibi ortalama olarak 1 milisaniyede 2000 ilişki döngüsü yapabilmektedir.

SONUÇ VE DEĞERLENDİRME

Bulut bilişim her ne kadar geçmişe dayanan, veri merkezlerinin ve büyük şirketlerin yıllardır belli ölçüde kullandığı bir teknoloji olmasına rağmen artan internet kullanımıyla son 10 yıl içerisinde çok popüler hale gelmiştir. Büyük teknoloji şirketlerinin buluta olan yatırımı ve yenilikleri kullanımın daha da artmasını sağlamıştır. Esasında bir telekomünikasyon terimi olan bulut, bir iletişim ağında büyük ölçekteki yapıları temsil eder. Buna karşın bulut bilişim, bu büyük ölçekteki yapı üzerinde çalışan uygulamalara, servislere, zaman ve mekana bağlı olmaksızın erişim imkanı sağlayan, ölçeklenebilir bir teknolojidir.

Bulut bilişim, bilgi teknolojileri servisleri için, internet tabanlı yeni bir dağıtım ve destek modelidir. Bu model, uzak cihazlara ve sunucu hizmetlerine internet üzerinden kolaylıkla erişim sağlama fikrinin bir ürünüdür. Tipik bulut bilişim, bir sunucu üzerinde tutulan veri ve yazılımlara web servisi ya da web tarayıcı benzeri yazılımlarla erişebilmeyi sağlar (Gruman ve Knorr, 2008).

Bulut bilişim, donanım bazında kullanıcıları soyutlayan bir teknolojidir. Bütün yönetim, bakım ve donanım yükü sağlayıcı tarafından yüklenilmektedir ve bu sayede kullanıcının omuzlarından büyük bir yük kalkmaktadır.

Bulut bilişim modelinde aynı zamanda “kullandığın kadar öde” prensibi yer almaktadır. Bu anlamda kullanıcılar ihtiyaçları doğrultusunda donanım ve servis kullanmakta, kullandığı hizmet kadar ücret ödemektedirler. Bu ücretler donanım hızı, internet bant genişliği, depolama alanı gibi kriterlere göre değişmektedir ve her sağlayıcının fiyatları farklı olabilmektedir. Ayrıca şirketler ekstra lisans ücreti ödemek zorunda kalmamaktadır. Uygulamaların performansları anlık olarak artmakta ve şirket maksimum verimi alabilmektedir.

Bulut bilişim sistemlerinde performansa etki edecek bazı kriterler aşağıda sıralanmıştır;

- **Güvenlik:** Güvenlik bulut performansında biraz tuhaf bir olgu olmasına rağmen ağ altyapısı için kanıtlanmış bir olgudur. Örneğin DDOS saldırıları ağ performansında

geniş bir etkisi vardır ve bu gerçekleşirse performansı düşürür ve sunucu cevap süresi üzerinde etkisi olmaktadır.

- **Kurtarma:** Buluttaki veri bir sorunla karşılaştığında ya da herhangi bir nedenden kaybedildiğinde, verinin geri getirilmesi için harcanan zaman performans üzerinde etkisi olmaktadır. Örneğin veri kurtarma çok zaman alırsa bulut performansı düşer ve müşteri memnuniyetsizliği artar.
- **Servis Düzeyi Anlaşması:** Bir kullanıcı bulut sisteminden faydalanmak istediğinde, sağlayıcı ve kullanıcı arasında, kullanıcının istekleri, sağlayıcının yapabilecekleri, ücretler, cezalar vb. içeren bir anlaşma imzalanmaktadır. Kullanıcı açısından bakıldığında optimal ve uygun zamanlı istekler için anlaşıldığında performans en iyi olacaktır.
- **Bant Genişliği:** Eğer ağdaki bant genişliği az olursa, performansta o kadar düşük olacaktır.
- **Depolama Kapasitesi:** Fiziksel hafıza performans için etkili olan bir kriterdir.
- **Hata Toleransı:** Bu kriterin bulut performansında önemli bir etkisi vardır. Örneğin bir veri merkezi yetersiz ve minimal servisi sağlıyorsa, bu performansı düşüren bir kriter olacaktır.
- **Erişilebilirlik:** Bir bulut sistemine ne kadar kolay ve hızlı erişilirse, performans o kadar yüksek olmaktadır.
- **Konum:** Veri merkezlerinin kullanıcılara olan uzaklık ve yakınlığı performansı etkileyen önemli kriterlerden biridir. Kullanıcı veri merkezine ne kadar uzak olursa, aldığı cevap süresi o kadar uzun olacak ve böylece performans düşecektir.

Bulut bilişim aynı zamanda mekan ve platformdan bağımsız olarak erişim altyapısı sunmakta ve istenilen boyutlarda ölçeklenebilmektedir. Ödenilen fiyata göre güçlü sunucular sunmaktadır. Bunun yanı sıra veri güvenliği, veri yedekleme, felaket yönetimi gibi olayları tamamen sağlayıcı yönetmekte, %99.5 gibi oranlarda erişim oranı sunmaktadır. Bu istenilen zamanda istenilen veriye ulaşılacağı anlamına gelmektedir. Kullanıcılar uygulamalarıyla ilgilenirken, diğer kalan işleri sağlayıcı yapmaktadır. Bulut teknolojisi kullanıcıların yükünü büyük oranlarda hafifletmektedir.

Bulut bilişimin karlı olduğunu gösteren karşılaştırma Şekil 31’da incelenmiştir.

	Dahili BT	Yönetilen Hizmetler	Bulut
Sermaye Yatırımı	\$40000	\$0	\$0
Kurulum Maliyetleri	\$10000	\$5000	\$1000
Aylık Hizmetler	\$0	\$4000	\$2400
Aylık İşgücü	\$3200	\$0	\$1000
3 Yıllık Maliyet	\$149000	\$129000	\$106000
Kazanılan Birikim	0%	13%	29%

Şekil 32: Bulut Bilişim ve Diğer Hizmetlerin Maliyet Karşılaştırması

Kaynak: (Rackspace)

Bulut bilişimin 2016 yılında pazar getirisi 21 milyon dolara yaklaşmış bulunmaktadır. 2018’e kadar bu getirinin 41 milyon dolara ulaşacağı tahmin edilmektedir. Yapılan incelemeler sonucu gözlenmektedirki gelecekte birçok firma, en azından belli ölçüde bulut bilişimden faydalanacaktır. Bulut bilişimin getirmiş olduğu kolaylıklar sayesinde dünya daha hızlı ve verimli uygulamalara sahne olacaktır.

Şirketlerin bulut teknoloji ile bünyelerinde barındırdığı yüksek maliyetli, yer kaplayan sunucuları kaldırılıp, uzaktaki veri merkezlerine taşımaktadır. Bu da yüksek maliyetlerden kurtulmanın yanında, farklı bir istihdamın oluşmasına neden olmaktadır. Bu da iş yapma şekillerinden personel istihdamına kadar, sektörü etkileyerek yeni iş kolları doğmasına yol açacaktır.

NoSQL veritabanları da bulut teknolojileri gibi temelleri geçmişe dayanan bir kavram olmasına karşın son 10-15 yıldır dünyadaki kullanımı büyük bir ivmeyle artmıştır. Dünyadaki internetin gelişimiyle beraber kullanıcıların artan talebi nedeniyle birçok teknoloji gibi NoSQL’ler de gelişmeler yaşamıştır. NoSQL kavramı daha da gelişmeye başlamış, birçok ürün ortaya çıkmıştır. Bütün bu gelişmeler, kullanıcıların uygulamalarının performanslarını artırmak, daha hızlı ve verimli uygulamalar ortaya çıkararak kullanıcılara daha iyi bir hizmet sunmaktır. Jablonski (2011:2), birçok şirketin organize edilmiş kendilerine özgü depolama sistemlerinin geliştirdiğini ve bunun NoSQL olarak adlandırıldığını belirtir.

NoSQL veritabanlarının gelişmesindeki en büyük etken, kullanıcıların ve sistemlerin oluşturduğu çok büyük verilerdir. Bu veriler genelde çok büyük boyutlarda ve anlamsız olmaktadır. Bu büyük ve anlamsız verileri analiz etmek, anlamlı veriler ortaya çıkarmak oldukça zordur. Bu noktada birçok NoSQL veritabanları devreye girmektedir. Verinin türü ve yapısına göre analiz kolaylığı değişmektedir ve veriye en uygun NoSQL veritabanını seçmek gerekmektedir. Bu anlamda dünya genelinde birçok NoSQL veritabanı yer almaktadır. Moniruzzaman ve Hossain (2013:2) belirttiği gibi, büyük verileri yönetebilmek için klasik ilişkili veritabanları yerine NoSQL, NewSQL gibi veritabanları vardır.

NoSQL, geliştiricileri klasik ilişkili veritabanlarının sağlayamadığı esnekliğe götürmektedir. Sağlamış olduğu şema bağımsız, JSON tabanlı basit yapıları ile çok büyük verilerin depolanmasını kolaylaştırmaktadırlar. Artan bu veriler kullanıcı deneyimi, uygulama performansı gibi kriterler için analiz edilebilmektedir.

NoSQL veritabanlarının avantajlar şöyle özetlenebilir;

- **Esnek Ölçeklendirme:** NoSQL veritabanları yeni nodlardan faydalanmak amacıyla transparan bir şekilde genişleyebilir tasarlanmıştır. Genelde az maliyetli donanım göz önünde bulundurularak tasarlanmıştır.
- **Büyük Veri:** NoSQL veritabanları ile, klasik veritabanlarına göre veri hacmi daha iyi yönetilmektedir.
- **DBA yok:** NoSQL veritabanları daha az yönetim gerektirecek şekilde tasarlanmıştır. Otomatik onarım, veri dağıtımı ve daha basit veri modelleri ile daha az yönetim ve bakım gereksinimleri sağlamaktadır.

Dünyada verinin nasıl arttığını gösteren grafik Şekil 32’da gösterilmiştir.



Şekil 33: Dünyadaki Veri Artış Grafiği

Kaynak: (Oracle, 2012)

Günbe gün artan veriler ile mücadele edebilmek için hemen her şirket NoSQL kullanımlarını artırmaya başlayacaktır. Gerek sistemlerde hızlı arama olsun, gerek kullanıcıların aktivitelerini takip etmek amaçlı olsun, gerek sistemin çekirdeğini NoSQL üzerine kurarak bir uygulama geliştirmek olsun, firmalar artan talepler yanıt vermek için NoSQL veritabanları kullanmaya başlayacak ya da kullanımlarını artıracaklardır.

Bulut teknolojilerinin de ilerlemesiyle bulut üzerinde NoSQL veritabanları kullanmak, hem hız ve performans sağlayacak, hem de maliyetleri ve iş gücünü azaltacaktır. NoSQL'lerin sunmuş olduğu gücü tam anlamıyla yakalamak için gereken donanım desteği bulut kullanılarak aşılabılır. Böylece herkes için kazanım olmuş olacaktır. Şirketler, ekstra donanım maliyet ve yönetiminden kurtulmuş olacaktır.

Yapılan örnek uygulama 1974 ve 2014 yılları arasında Türkiye'de gerçekleşmiş olan terör olaylarının ampirik incelemesine odaklanmıştır. Terör olayları verileri Maryland Üniversitesi Global Terrorism Database'inden elde edilmiş olup modelleme için Neo4j veri tabanı kullanılmıştır. Veriler CSV formatından okunarak veri tabanı Node'ları oluşturulmuş ve veri tabanına aktarılmıştır.

Bu uygulamada terör olaylarının görsel olarak incelebilmesi ve yorumlanarak önlemlerin alınabileceği bir modelleme önerilmiştir. Model üzerinde inceleme amaçlı 3 farklı sorgulama yapılmıştır. Bu sorgular;

- 2014 yılında Ankara’da ölümle sonuçlanmış eylemler,
- 2005 ve 2014 yılları arasında Hakkâri’de Bombalama/Patlama tipinde gerçekleştirilen terör eylemleri,
- 1974 ve 2014 yılları arasında bombalama/patlama tipindeki saldırıların en çok gerçekleştiği şehirler

Sonuçlarını bulmaya yönelik olmuştur. Özellikle, yapılan 1974 ve 2014 yılları arasında bombalama/patlama tipindeki saldırıların en çok gerçekleştiği şehirler sorgulaması sonucunda saldırıların en çok İstanbul şehrine yönelik olduğu gözlemlenmiştir.

Türkiye Büyük Millet Meclisi’nin 2013 24.dönem 3. Yasama yılında yayınlamış olduğu “Terör ve Şiddet Olayları Kapsamında Yaşam Hakkı İhlallerini İnceleme Raporu’na göre 1993-2011 yılları arasında silahlı saldırı sonucu ölüm verileri Tablo 10’da verilmiştir.

Tablo 10
1993-2011 Yılları Arasında Ölüm Verileri

YIL	SİLAHLI ÇATIŞMALARDA ÖLENLER	SİVİLLERE YÖNELİK ÖLDÜRMELER	KARA MAYINLARI- SERBEST PATLAYICILARIN SEBEP OLDUĞU ÖLÜMLER	ASKER-POLİS ŞÜPHELİ İNTİHARLARI
1993	3333	807	-	-
1994	5000	458	-	-
1995	3894	230	-	-
1996	2859	119	-	-
1997	2514	151	-	-
1998	1718	91	-	-
1999	857	138	-	-
2000	147	52	-	-
2001	92	30	-	-
2002	30	7	35	-
2003	104	6	20	18
2004	240	28	59	18
2005	499	39	70	17
2006	345	82	40	21
2007	424	27	14	36
2008	432	22	28	24
2009	141	36	22	34
2010	244	51	5	47
2011	318	21	-	31
TOPLAM	22.971	2.295	293	246

Kaynak: (TBMM, 2013:56)

Tablodaki verilere göre; 1993-2011 yılları arasındaki 19 yıllık dönemde 22.971’i Silahlı Çatışmalarda Ölüm, 2.295’i Sivillere Yönelik Öldürmeler, 293’ü Kara Mayınları ve Serbest Patlayıcıların Sebep Olduğu Ölüm ve 246’sı da Asker-Polis Şüpheli

İntiharı olmak üzere toplam 25.805 kişi terörden kaynaklı yaşam hakkı ihlaline uğramıştır (TBMM, 2013:57).

Yine aynı raporda yayınlanan 1984-2012 yılları arasında jandarma sorumluluk bölgesinde yaşamını yitiren sivil verileri Tablo 11’de gösterilmiştir.

Tablo 11
1984-2012 Arasında Jandarma Bölgesinde Yaşamını Yitiren Siviller

	YIL	HAYATINI KAYBEDEN SIVİL SAYISI
1	1984	20
2	1985	67
3	1986	57
4	1987	217
5	1988	91
6	1989	160
7	1990	160
8	1991	127
9	1992	487
10	1993	932
11	1994	682
12	1995	292
13	1996	130
14	1997	148
15	1998	71
15	1999	78
16	2000	27
17	2001	11
18	2002	11
19	2003	10
20	2004	16
21	2005	18
22	2006	18
23	2007	20
24	2008	29
25	2009	18
26	2010	21
27	2011	17
28	2012	6
	TOPLAM	3.924

Kaynak: (TBMM, 2013:64)

Tablo incelendiğinde 1984 ile 1991 arasında 20 ile 217 aralığında değişen rakamlarda sivil kayıpları yaşanmışken 1991-1996 arasında bu sayının önemli artış gösterdiği ve bu rakamın 1993 yılında 932’ye kadar yükseldiği daha sonra 2000 ve izleyen yıllarda 6-29 aralığında seyredecek seviyelere indiği, 1984 ile 2012 arasındaki 28 yılda toplam 3.924 sivilin Jandarma bölgesinde terör olayları nedeniyle yaşamını yitirdiği görülmektedir (TBMM, 2013:65).

Uygulanan model verileri ve TBMM verileri incelendiğinde terör eylemlerinin 70’li yıllarda başladığı, 90’lı yıllarda önemli derecede artış gösterdiği ve günümüze kadar devam ettiği gözlemlenmektedir.

Bu modelleme ile gerçekleştirilen terör eylemlerinin görsel bir yapısının ortaya çıkarılması, model üzerinde güvenlik güçleri ve ya ilgili birimlerin araştırma yaparak gerekli önlemlerin alınmasının sağlanması amaçlanmıştır. Modeldeki verilerden aşağıdaki bazı bilgilere ulaşılabilmektedir;

- Hangi bölge ve ya hangi şehirde saldırıların en çok olduğu,
- Gerçekleştirilen bu saldırıların en çok hangi türde olduğu,
- Bu saldırılar ile en çok hedef alınan kişi ve ya kurumların kim olduğu,
- Bu saldırılarda en çok hangi tür silahın kullanıldı,
- Saldırıların yıl içerisinde en çok hangi ayda yoğunlaştığı, buna bağlı olarak hangi mevsimde yoğunlaştığı
- Hangi tür saldırıların en çok hangi grup tarafından hangi bölge/şehir’lerde gerçekleştirildiği,
- En çok can ve varsa mal kaybının hangi tür saldırılarda olduğu

yapılan sorgulamalar daha farklı bilgilere ulaşmak amacıyla genişletilebilmektedir. Bu sayede hedef, kullanılan silahlar, şehirler gibi varlıklar için alınan önlemlerin artırılması amaçlanmıştır. Modellenen veri tabanını bir kullanıcı arayüzü üzerinden sunularak cevabının bulunması istene sorgular sürükle bırak yardımıyla oluşturulabilir ve böylece sorgulama dili bilmeyen bir kullanıcı tarafından rahatlıkla yapılması ileriki aşamalarda sağlanabilir. Bu sayede gerçekleştirilecek muhtemel eylemler için önlenm alınması daha rahat olacaktır.

Uygulamanın diğer bir amacı da büyük verilerin, klasik ilişkili veri tabanları yerine farklı yapıları olan NoSQL veri tabanları kullanarak daha iyi analiz edilebileceğini göstermektir. Kullandığımız Neo4j veri tabanı verileri graph’lar üzerinde barındırdığından, sonuçlara görsel olarak ve akılda kalıcı şekilde ulaşmamızı sağlamaktadır.

Verilerin detaylı bir şekilde elde edilmesi, sistemin saęlıęı aısından en önemli kısıtlamadır. Gerekleřtirilen eylemlerinin biroęunda gerekleřtięi yerin koordinat bilgisi bulunmaktadır. Koordinat bilgisi bulunmayan kayıt sayısı 863 adettir fakat bunların biroęu eski tarihli gerekleřmiř eylemlerdir. Bu nedenle řehirlerde gerekleřebilecek olası eylemler iin risk haritası ıkarmak %100 olmasa da byk oranda mmkndr. Ancak adres bilgilerinin de sistem zerinde tamamlanarak yapılacak sorgulamalar ile daha iyi sonu alınması saęlanabilir.

Modellenen rnek uygulama Trkiye’de gerekleřmiř terr olaylarına odaklanmıřtır. Bu olayların sayısı 3144’tr. Fakat veri kaynaęı incelendięinde dnya genelinde yaklařık 150 bin terr olayı olduęu gzlemlenmektedir. Bu sayıdan yola ıkarak sistemdeki koordinat, olayın gerekleřtięi yer, saldırı tipi ve hedefi, zet bilgi gibi birok iře yarar bilgi kullanarak bir makine ęrenmesi gerekleřtirilebilir. Bu akıllı sistem daha nce gerekleřmiř bu olayları ęrenerek ileride gerekleřebilecek olaylar iin ıkarımlarda bulunabilir. Daha sonra bu sistem ile bir veri akıřı entegre edilip muhtemel gerekleřecek bir terr eylemini nceden haber verebilecek, akademik baęlamda arge yapılabilecek bir karar destek sistemi gerekleřtirilebilir.

Yapılan sorgulamalar ve TBMM’nin verileri gz nnde bulundurulduęunda, gvenlik glerinin terr olaylarına karřı nlem alması ve can kaybını nlemesi, terr olaylarının gerekleřmeden engellenmesi byk nem arz etmektedir. Bu sistem ile gvenlik glerinin, gerekleřmiř olayları incelemesi, bu olayları analiz etmesi, gerekleřtirilecek olan makine ęrenmesi sistemi ile olayların gerekleřmeden nlem alması mmkn olacaktır.

KAYNAKÇA

Kitaplar

- ALBAYRAK, Ali Sait, KOLTAN YILMAZ, Şebnem (2009), “Veri Madenciliği: Karar Ağacı Algoritmaları ve İMKB Verileri Üzerine Bir Uygulama”, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi Y.2009, C.14, S.1 s.31-52
- BACHMAN, Michal, MISQUITTA, Luanne, GEORGE, Adam, BICKERS, Vince, GIERKE, Oliver, HUNGER, Michael (2016, Versiyon 4.1.2 Release), “Good Relationships - The Spring Data Neo4j Guide Book”
- CODD, E. F. (1970). “A relational model of data for large shared data banks. Communications of the ACM”, 13(6), 377-387.
- EIFREM, Emil, WEBBER, Jim, ROBINSON, Ian (2015 Neo Technology), “Graph Databases – New Opportunities For Connected Data”, s. 1-210
- FOWLER, Adam (2015), “NoSQL for Dummies”, ISBN 978-1-118-90574-6 (pbk), ISBN 978-1-118-90562-3 (epub), ISBN 978-1-118-90578-4 (epdf)
- GARFINKEL, S. L., and ABELSON, H.,(1999), Architects of the information society: Thirty-five years of the laboratory for computer science at MIT. Cambridge, Massachusetts, U.S.A: MIT Press
- HEWITT, Eben (2010), “Cassandra: the Definitive Guide”, O'Reilly Media, ISBN: 978-1-449-39041-9
- KRISHNAN, S. (2010). “Programming Windows Azure”. O'Reilly Media, 1, 368p. United States of America
- KULKARNI, Gurudatt, SUTAR, Ramesh ve GAMBHIR, Jayant (2012), “Cloud Computing-Infrastructure as Service Amazon EC2”, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 1, Jan-Feb 2012, pp.117-125 Bigtable: A Distributed

Storage System for Structured Data Fay Chang, Jeffrey Dean, Sanjay Ghemawat,
Wilson C. Hsieh, Deborah A. Wallach

TONG, Zachary ve GORMLEY, Clinton (2015), “Elasticsearch: The Definitive Guide”,
O’Reilly Media, ISBN: 978-1-449-35854-9

ZHAO, L., SAKR, S., LIU, A., BOUGUETTAYA, A., 2014, “Cloud Data
Management” XIX, 202p. 86 illus., 50illus. in color, Hardcover

Süreli Yayınlar

A Mongo White Paper (2015) , “Top 5 Considerations When Evaluating NoSQL
Databases”, https://s3.amazonaws.com/info-mongodb-com/10gen_Top_5_NoSQL_Considerations.pdf

ABM, Moniruzzaman ve SYED, Akhter Hossain (2013). “Nosql database: New era of
databases for big data analytics-classification, characteristics and comparison”,
International Journal of Database Theory & Application, 6(4),
http://www.sersc.org/journals/IJDTA/vol6_no4/1.pdf

ABRAMOVA, V., BERNADINO, J., FURTADO, P., (2014), Which NoSQL Database?
A Performance Overview, Polytechnic Institute of Coimbra - ISEC / CISUC,
Coimbra, Portugal, Open Journal of Databases (OJDB) Volume 1, Issue 2, 2014
ISSN 2199-3459

Amazon Web Services Inc. (2013), “Amazon Simple DB Developer Guide”, API
Version 2009-04-15

BTK (BİLGİ TEKNOLOJİLERİ VE İLETİŞİM KURUMU) (2013), Kasım 2013, Bulut
Bilişim, Ankara,
https://www.btk.gov.tr/File/?path=ROOT%2F1%2FDocuments%2FSayfalar%2FArastirma_Raporlari%2FBulut_Bilisim.pdf

- CHANDRAN Shena Prabha ve ANGEPAT Mridula, 2010, “Cloud Computing: Analysing the risks involved in cloud computing environments”, School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
- CHANG, Fay, DEAN, Jeffrey, GHEMAWAT, Sanjay, C.HSIEH, Wilson, A.WALLACH, Deborah, BURROWS, Mike, CHANDRA, tushar, FIKES ve Andrew, E. GRUBER, Robert (2006), “Bigtable: A Distributed Storage System for Structured Data”
- CHAPPELL, David (2012), “Windows Azure Data Management Choosing The Right Technology”, http://www.davidchappell.com/writing/white_papers/Windows-Azure-Data-Mgmt--Choosing-the-Right-Technology--Chappell-v1.0.pdf
- CHAPPELL, David (2008), “Introducing the Azure Services Platform - An Early Look At the Windows Azure, .NET Services, and Live Services”, https://pplware.sapo.pt/downloads/azure_services_platform.pdf
- DECANDIA, Giuseppe, HASTORU, Deniz, JAMPANI, Madan, KAKULAPATI, Gunavardhan, LAKSHMAN, Avinash, PILCHIN, Alex, SIVASUBRAMANIAN, Swaminathan, VOSSHALL, Peter ve VOGELS, Werner, Dynamo: “Amazon’s Highly Available Key-value Store, Amazon”
- DEMİRCİOĞLU, Bahadır (2016), Nisan 2016, “Sanallaştırma Nedir”
- GONG, C., LIU, J., ZHANG. Q., CHEN, H., and GONG, Z., (2010), The characteristics of cloud computing. 39th International Conference on Parallel Processing Workshops. 13-16 September 2010, San Diego, California, U.S.A, pp. 275-279.
- GORELIK, Euhene (2013), “Cloud Computing Models”, Composite Information Systems Laboratory (CISL) Sloan School of Management, Room E62-422 Massachusetts Institute of Technology Cambridge, MA 02142
- GRUMAN, G., KNORR, E. (2008). What cloud computing really means. <http://www.infoworld.com/d/cloud-computing/what-cloud-computingreallymeans-031>.

- HUDDAR, M.G., RAMANNAVAR, M.M. (2013). A Survey on Big Data Analytical Tools. International Journal of Latest Trends in Engineering & Technology. Special Issue, IDEAS.
- IDC (2010), Haziran 2010, Worldwide and Regional Public IT Cloud Services 2010-2014 Forecast
- IBM (2015), Mart 2015, “Why NoSQL? Your Database Options In The New Non-relational World”, Analytics White Paper
- JABLONSKI, Stefan ve HECHT, Robin (2011), “NoSQL Evaluation-A Use Case Oriented Survey”
- JEFFREY, F.Rayport ve HEYWARD, Andrew (2009), 20 Mart 2009, “Envisioning the Cloud: the Next Computing Paradigm”
- KHANGHAHI, Niloofar ve RAVANMEHR, Reza (2013), Ekim 2013, “CLOUD COMPUTING PERFORMANCE EVALUATION: ISSUES AND CHALLENGES”, Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran
- KRIUSHANT M., AROCKIAM L., JUSTY MIROBU G (2013), Temmuz 2013, “Auto Scaling in Cloud Computing : An Overview”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 7
- LICKLIDER, J. C. R., and TAYLOR, R. W., (1968), The computer as a communication device, Systems Research Center.
<http://sloan.stanford.edu/mousesite/Secondary/Licklider.pdf>.
- NWOBODO, Ikechukwu (2015), “A Comparison of Cloud Computing Platforms”, School of Architecture Computing and Engineering University of East London, International Conference on Circuits and Systems (CAS 2015)
- TUTORIALS POINT (2015), “CouchDB-No Document Store”,
https://www.tutorialspoint.com/couchdb/couchdb_tutorial.pdf

TÜRKİYE BÜYÜK MİLLET MECLİSİ (2013), 13 Şubat 2013, 24. Dönem 3. Yasamı Yılı, “TERÖR VE ŞİDDET OLAYLARI KAPSAMINDA YAŞAM HAKKI İHLALLERİNİ İNCELEME RAPORU”

ÖZTÜRK, Serdar, ATMACA, Hatice Ediz (2017), “İlişkisel ve İlişkisel Olmayan (NoSQL) Veri Tabanı Sistemleri Mimari Performansının Yönetim Bilişim Sistemleri Kapsamında İncelenmesi”, BİLİŞİM TEKNOLOJİLERİ DERGİSİ, CİLT: 10, SAYI: 2, NİSAN 2017, Gazi Üniversitesi, DOI: 10.17671/gazibtd.309303

VARIA, Jinesh ve MATHEW Sajee (2014), Ocak 2014, “Overview of Amazon Web Services”

Tezler

ATAŞ, Gültekin (2013), “Performance Evaluations of Cloud Computing Platforms”, Bahçeşehir Üniversitesi, İstanbul

AYDEMİR, Fatir (2016), Haziran 2016, “Sınır Güvenliği İçin Büyük Veri Teknil ve Teknolojileri İle Boru Hattı Tasarımı”, Bilgisayar Mühendisliği Anabilim Dalı Fen Bilimleri Enstitüsü Gazi Üniversitesi

BITCHI, Bashir Yusuf (2014), 6 Haziran 2014, “Efficient Resource Management Framework In Cloud Computing”, Yaşar Üniversitesi Bilgisayar Mühendisliği, İzmir

BEDİROĞLU, Şevket (2013), “Web Haritalarının bulut Bilişim İle Yayınlanması: Trabzon İli Örneği”, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Harita Mühendisliği Anabilimdalı, Trabzon

ÇATAK, Ferhat Özgür (2014), “Bulut Bilişim Sistemlerin Eşle/İndirge Yöntemi Kullanılarak Veri Madenciliği Yazılım Çatısının Geliştirilmesi”, İstanbul Üniversitesi Fen Bilimleri Enstitüsü Enformatik Anabilim Dalı

- HENRICSSON, Robin (2011), 13 Haziran 2011, “Document Oriented NoSQL Databases A Comparisons of Performance in MongoDB and CouchDB using a Python Interface”, Blekinge Institute of Technology
- LAMLLARI, Rilinda (2013), 27 Haziran 2013, “Extending a Methodology for Migration of the Database Layer to the Cloud Considering Relational Database Schema Migration to NoSQL”, Master Thesis No. 3460, Institute of Architecture of Application Systems University of Stuttgart
- MUHAMMAD, Yousaf (2011), “Evaluation and Implementation of Distributed NoSQL Database for MMO Gaming Environment”, Department Of Information Technology Uppsala University
- OREND, KAI (2010), “Analysis and Classification of NoSQL Databases and Evaluation of Their Ability of Replace an Object-relational Persistence Layer”, Technische Universität München
- NÄSHOLM, Petter (2012), “Extracting Data from NoSQL Database A Step Towards Interactive Visual Analysis of NoSQL Data”, Department of Computer Science and Engineering Chalmers University of Technology
- NICOLETA, C.Brad (2012), “Data De-Duplication In NoSQL Databases”, Department of Computer Science University of Saskatchewan
- ROIJACKERS, John (2012), 11 Mayıs 2012, “Bridging SQL and NoSQL”, Department of Mathematics and Computer Science Eindhoven University of Technology
- SEVLİ, Onur (2011), “Bulut Bilişim ve Eğitim alanında Örnek Uygulama”, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, Isparta
- SHERMIN, Motahera (2013), “An Access Control Model for NoSQL Databases”, The University of Western Ontario

SUTER, Rico (2012), Ocak 2012, “MongoDB-An Introduction and Performance Analysis”, Seminar Thesis, Master of Science in Engineering, Major Software and Systems HSR Hochschule für Technik Rapperswil

UZUNBAYIR, Serhat (2015), “ A Comparison Between Relational Database Models And NoSQL Trends On Big Data Design Challenges Using A Social Shopping Application”, Natural and Applied Sciences of Izmir University of Economics

Diğer kaynaklar

30 Years of Accumulation: A Timeline of Cloud Computing (2013), 30 Mart 2013, <https://gcn.com/articles/2013/05/30/gcn30-timeline-cloud.aspx>

Amazon DynamoDB, <https://aws.amazon.com/dynamodb/>

Amazon, <https://aws.amazon.com/>

BJORNSEN, Zach (2015) , 29 Aralık 2015, “AWS S3 vs Google Cloud vs Azure: Cloud Storage Performance”, <http://blog.zachbjornsen.com/2015/12/29/cloud-storage-performance.html>

BLODGET, Henry (2013), “The Number Of PCs”, <http://www.businessinsider.com/number-of-smartphones-tablets-pcs-2013-12>

BORT, Julie (2013), 20 Nisan 2013, The 10 Most Important Companies In Cloud Computing, <http://www.businessinsider.com/10-most-important-in-cloud-computing-2013-4>

BRUGGEN, Rik Van, (2015), 21 Eylül 2015, “Experimenting with A POLE, the Global Terrorism Database, and Neo4j”, <http://blog.bruggen.com/search/label/global%20terrorism%20database>

COLOMBUS, Louis (2016), 4 Şubat 2016, The Best Cloud Computing Companies And CEOs To Work For In 2016, <http://www.forbes.com/sites/louiscolumbus/2016/02/04/the-best-cloud-computing-%20companies-and-ceos-to-work-for-in-2016/#7c86882a581b>

“Cypher is the declarative query language for Neo4j, Cypher is the declarative query language for Neo4j, the world’s leading graph database. the world’s leading graph database” , <http://neo4j.com/docs/pdf/neo4j-cypher-refcard-stable.pdf>

DOLLARD, Morgan (2015), “Bringing you more flexibility and better Cloud Networking performance, GA of HTTPS Load Balancing and Akamai joins CDN , Interconnect”, <https://cloudplatform.googleblog.com/2015/11/bringing-you-more-flexibility-and-better-Cloud-Networking-performance-GA-of-HTTPS-Load-Balancing-and-Akamai-joins-CDN-Interconnect.html>

DORMAN Scot, What is Windows Azure?
<https://stackoverflow.com/questions/240455/what-is-windows-azuredowntime-than-main-cloud-rivals>

DUDLEY R.J. (2016), 28 Mart 2016, Setting Up Neo4j on Azure VM,
<http://rjdudley.com/setting-up-neo4j-on-azure-vm/>

FLINDERS, Karl (2015), 19 Ocak 2015, “Microsoft Azure Had More Downtime in 2014 Than Main Cloud Rivals”,
<http://www.computerweekly.com/news/2240238379/Microsoft-Azure-had-more-downtime-than-main-cloud-rivals>

FORREST, Conner (2014), 12 Eylül 2014, “The Google Cloud Platform comprises many of Google's top tools for developers. Here are 10 things you might not know about it.”, <http://www.techrepublic.com/article/the-google-cloud-platform-10-things-you-need-to-know/>

GENTZ, Mimi (2016), “Introduction to DocumentDB: A NoSQL JSON Database”,
<https://azure.microsoft.com/en-us/documentation/articles/documentdb-introduction/>

Google App Engine, <https://cloud.google.com/appengine/>

GRUMAN, G., KNORR, E., 2008. “What cloud computing really means”.

<http://www.infoworld.com/d/cloud-computing/what-cloud-computingreallymeans-031>

GUTHRIE, Scott (2012), 28 Ağustos 2012, “Announcing Windows Azure Mobile Services”, <https://weblogs.asp.net/scottgu/announcing-windows-azure-mobile-services>

Introduction about cloud computing, <http://www.salesforcetutorial.com/introduction-to-cloud-computing/>

Kodcu.com (2014), Netflix Mühendisi Nezih Yiğitbaşı’ndan Big Data Webieri, <https://kodcu.com/2014/12/netflix-muhendisi-nezih-yigitbasidan-big-data-webineri/>

LAMMER, Frank (2016), 13 Ocak 2016, Is Platfowm as a Service Dead?, <https://blog.fortrabbbit.com/cloudscapes-rerevisited>

MASSIMO, Bensi (2016), 17 Şubat 2016, <http://techblog.suitsupply.com/2016/02/17/cloud-computing-using-microsoft-azure-part-1-by-massimo-bensi/>

Microsoft Azure, <https://azure.microsoft.com/tr-tr/>

MongoDB, <https://www.mongodb.com/>

NEO4J, <https://neo4j.com/>

NETMERA (2016), 29 Nisan 2016 , <http://www.slideshare.net/NetmeraMarketing/netmera-slideshare-vs>

NETO, Maximilliano Destefani (2014), “A brief history of cloud computing”, <http://www.thoughtsoncloud.com/2014/03/a-brief-history-of-cloud-computing/>

PANKAJ Sonawane, Big Data and E-Commerce, <http://www.slideshare.net/PankajSonawane11/clipboards/1>

PERRY, G., (2008), 28 Şubat 2008, How cloud & utility computing are different,
<http://gigaom.com/2008/02/28/how-cloud-utility-computing-are-different/>

POPESCU, Alex (2010), “ Presentation: An Interesting NoSQL categorization”,
<http://nosql.mypopescu.com/post/396337069/presentation-nosql-codemash-an-interesting>

QUICKBASE, A TimeLine of Database History,
<http://www.quickbase.com/articles/timeline-of-database-history>

RACKSPACE, Cloudonomics: The Economics of Cloud Computing,
<https://support.rackspace.com/white-paper/cloudonomics-the-economics-of-cloud-computing/>

Radar Networks and Nova Spivack, 2007. Semantics of Connections.
<http://www.radarnetworks.com>

RODRIGUEZ, Marko(2011), 5 Aralık 2011, Database Zone, MySQL vs. Neo4j on a Large-Scale Graph Traversal, <https://dzone.com/articles/mysql-vs-neo4j-large-scale>

SALESFORCE, <https://www.salesforce.com/eu/>

SCOFIELD, Ben (2010): “NoSQL – Death to Relational Databases(?)” Presentation at the CodeMash conference in Sandusky (Ohio).
<http://www.slideshare.net/bscofield/nosql-codemash-2010>

Semantic Technologies and Linked Data Foundations
<http://sites.linkeddata.center/help/devop/training/ebc1p1>

Seven Incredible Benefits of Cloud Computing, 21 Şubat 2015,
<http://www.zoombd24.com/seven-incredible-benefits-cloud-computing/>

Suman Singh, Cloud Computing for Beginners
<http://pt.slideshare.net/SumanSingh3/cloud-computing-for-beginners>

Tekzan, <http://www.tekzan.com/>

TÜİK, Türkiye İstatistik Kurumu (2016), 18 Ağustos 2016 10:00, Sayı 21779,
“Hanehalkı Bilişim Teknolojileri Kullanım Araştırması,
<http://www.tuik.gov.tr/PreHaberBultenleri.do?id=21779>

University Of Maryland (2015), GLOBAL TERRORISM DATABASE,
<http://www.start.umd.edu/gtd/>

Azure Yapılandırma Kaynakları

<http://blog.keithics.com/mongodb-and-node-endpoints-with-new-azure-portal/>

<http://webbercross.azurewebsites.net/manually-installing-a-neo4j-database-on-an-azure-vm/>

<https://neo4j.com/developer/kb/how-do-i-enable-remote-https-access-with-neo4j-30x/>

<https://stackoverflow.com/questions/28220526/remote-connection-to-neo4j-server>

<https://stackoverflow.com/questions/41110305/connecting-to-neo4j-on-ec2>

EKLER

EK-1

MODEL OLUŞTURULMASINDA KULLANILAN KOMUTLAR

// Global Terrorism Veritabanı yükleme

//Yüklemeye devam edebilmek için veillerin bulunduğu CSV dosyasını indirip lokal bir klasöre koymak gerekli

//file:/gtdb_transformtr.csv and then to a local file.

//Ülkeleri yükleme

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct  
toInt(csv.ulke) as ulke, csv.ulke_metin as ulke_metin create (c:Ulke {id: ulke, name: ulke_metin});
```

```
create index on :Ulke(id);
```

schema await

//Bölgeleri yükleme

```
load csv with headers from "file:/gtdb_transformtr.csv" as csv fieldterminator ';' with distinct  
toInt(csv.bolge) as bolge, csv.bolge_metin as bolge_metin, toInt(csv.ulke) as ulke match (c:Ulke {id:  
ulke}) merge (r:Bolge {id: bolge, name: bolge_metin}) merge (c)-[:PARCASI]->(r);
```

```
create index on :Bolge(id);
```

```
create index on :Eyalet(name);
```

schema await

//Eyaletleri ekleme

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv
```

```
" as csv fieldterminator ';' with csv.eyalet as eyalet, toInt(csv.ulke) as ulke where eyalet is not null match  
(c:Ulke {id: ulke}) merge (ps:Eyalet {name: eyalet})-[:PARCASI]->(c);
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv  
fieldterminator ';' with csv.eyalet as eyalet, toInt(csv.ulke) as ulke where eyalet is null match (c:Ulke {id:  
ulke}) merge (ps:Eyalet {name: "Bilinmeyen"})-[:PARCASI]->(c);
```

//Şehirleri eyaletler ile bağlantılı ekleme

```
create index on :Sehir(name);
```

schema await

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with csv.eyalet as eyalet, toInt(csv.ulke) as ulke, csv.sehir as sehir where eyalet is not
null match (ps:Eyalet {name: eyalet})-[:PARCASI]->(c:Ulke {id: ulke}) merge (ci:Sehir {name: sehir})-
[:PARCASI]->(ps);
```

```
//Eyaletsiz şehirleri ekleme
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with csv.eyalet as eyalet, toInt(csv.ulke) as ulke, csv.sehir as sehir where eyalet is null
match (ps:Eyalet {name: "Bilinmeyen"})-[:PARCASI]->(c:Ulke {id: ulke}) merge (ci:Sehir {name:
sehir})-[:PARCASI]->(ps);
```

```
//Saldırı tiplerini ekleme
```

```
create index on :SaldiriTipi(id);
```

```
schema await
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct toInt(csv.saldiritipi1) as saldiritipi, csv.saldiritipi1_metin as
saldiritipi_metin where saldiritipi is not null merge (at:SaldiriTipi {id: saldiritipi, name:
saldiritipi_metin});
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct toInt(csv.saldiritipi2) as saldiritipi, csv.saldiritipi2_metin as
saldiritipi_metin where saldiritipi is not null merge (at:SaldiriTipi {id: saldiritipi, name:
saldiritipi_metin});
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct toInt(csv.saldiritipi3) as saldiritipi, csv.saldiritipi3_metin as
saldiritipi_metin where saldiritipi is not null merge (at:SaldiriTipi {id: saldiritipi, name:
saldiritipi_metin});
```

```
//Hedef tiplerini ekleme
```

```
create index on :HedefTipi(id);
```

```
schema await
```

```
create index on :HedefTipi(name);
```

```
schema await
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.hedeftipi1) as hedeftipi, csv.hedeftipi1_metin as hedeftipi_metin where hedeftipi is not null
merge (at:HedefTipi {id: hedeftipi, name: hedeftipi_metin});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.hedeftipi2) as hedeftipi, csv.hedeftipi2_metin as hedeftipi_metin where hedeftipi is not null
merge (at:HedefTipi {id: hedeftipi, name: hedeftipi_metin});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.hedef tipi3) as hedef tipi, csv.hedef tipi3_metin as hedef tipi_metin where hedef tipi is not null
merge (at:Hedef Tipi {id: hedef tipi, name: hedef tipi_metin});
```

```
//Hedefleri oluşturma
```

```
create index on :Hedef(name);
```

```
schema await
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct csv.hedef1 as hedef where hedef is not null merge (t:Hedef {name:
hedef});
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct csv.hedef2 as hedef where hedef is not null merge (t:Hedef {name:
hedef});
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct csv.hedef3 as hedef where hedef is not null merge (t:Hedef {name:
hedef});
```

```
//Hedefleri hedef tipleri ile ilişkilendirme
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct csv.hedef1 as hedef, toInt(csv.hedef tipi1) as hedef tipi where hedef is not
null match (tt:Hedef Tipi {id: hedef tipi}), (t:Hedef {name: hedef}) merge (t-[:TURU]->(tt);
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct csv.hedef2 as hedef, toInt(csv.hedef tipi2) as hedef tipi where hedef is not
null match (tt:Hedef Tipi {id: hedef tipi}), (t:Hedef {name: hedef}) merge (t-[:TURU]->(tt);
```

```
using periodic commit 2500 load csv with headers from "file:/gtdb_transformtr.csv " as csv
fieldterminator ';' with distinct csv.hedef3 as hedef, toInt(csv.hedef tipi3) as hedef tipi where hedef is not
null match (tt:Hedef Tipi {id: hedef tipi}), (t:Hedef {name: hedef}) merge (t-[:TURU]->(tt);
```

```
//Kurumları oluşturma
```

```
create index on :Kurum(name);
```

```
schema await
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct csv.kurum1
as kurum where kurum is not null merge (c:Kurum {name: kurum});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct csv.kurum2
as kurum where kurum is not null merge (c:Kurum {name: kurum});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct csv.kurum3
as kurum where kurum is not null merge (c:Kurum {name: kurum});
```

```
//Kurumları hedefler ile ilişkilendirme
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct csv.kurum1
as kurum, csv.hedef1 as hedef where kurum is not null and hedef is not null match (t:Hedef {name:
hedef}), (c:Kurum {name: kurum}) merge (c)-[:ILISKILI]->(t);
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct csv.kurum2
as kurum, csv.hedef2 as hedef where kurum is not null and hedef is not null match (t:Hedef {name:
hedef}), (c:Kurum {name: kurum}) merge (c)-[:ILISKILI]->(t);
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct csv.kurum3
as kurum, csv.hedef3 as hedef where kurum is not null and hedef is not null match (t:Hedef {name:
hedef}), (c:Kurum {name: kurum}) merge (c)-[:ILISKILI]->(t);
```

//Hedefleri ülkeler ile ilişkilendirme

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.uyruk1) as uyruk, csv.hedef1 as hedef where uyruk is not null and hedef is not null match
(c:Ulke {id: uyruk}), (t:Hedef {name: hedef}) merge (t)-[:BURADAN]->(c);
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.uyruk2) as uyruk, csv.hedef2 as hedef where uyruk is not null and hedef is not null match
(c:Ulke {id: uyruk}), (t:Hedef {name: hedef}) merge (t)-[:BURADAN]->(c);
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.uyruk3) as uyruk, csv.hedef3 as hedef where uyruk is not null and hedef is not null match
(c:Ulke {id: uyruk}), (t:Hedef {name: hedef}) merge (t)-[:BURADAN]->(c);
```

//Grupları yükleme

```
create index on :Grup(name);
```

schema await

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct csv.grupadi
as grup where grup is not null merge (g:Grup {name: grup});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
csv.grupadi2 as grup where grup is not null merge (g:Grup {name: grup});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
csv.grupadi3 as grup where grup is not null merge (g:Grup {name: grup});
```

//Silah tiplerini yükleme

```
create index on :Silah(id);
```

schema await

```
create index on :Silah(name);
```

schema await

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.silahtipi1) as silahtipi, csv.silahtipi1_metin as silahtipi_metin where silahtipi is not null merge
(w:Silah {id: silahtipi, name: silahtipi_metin});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.silahtipi2) as silahtipi, csv.silahtipi2_metin as silahtipi_metin where silahtipi is not null merge
(w:Silah {id: silahtipi, name: silahtipi_metin});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.silahtipi3) as silahtipi, csv.silahtipi3_metin as silahtipi_metin where silahtipi is not null merge
(w:Silah {id: silahtipi, name: silahtipi_metin});
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with distinct
toInt(csv.silahtipi4) as silahtipi, csv.silahtipi4_metin as silahtipi_metin where silahtipi is not null merge
(w:Silah {id: silahtipi, name: silahtipi_metin});
```

```
//Eylemleri yükleme
```

```
create index on :Eylem(id);
```

```
schema await
```

```
using periodic commit load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';'
create (e:Eylem {id: toInt(csv.eylemid), year: toInt(csv.iyear), month: toInt(csv.imonth), day:
toInt(csv.iday) , latitude: csv.latitude, longitude: csv.longitude, specificity: csv.specificity, vicinity:
csv.vicinity, location: csv.location, summary: csv.summary, crit1: toInt(csv.crit1), crit2: toInt(csv.crit2),
crit3: toInt(csv.crit3), doubt: toInt(csv.doubtterr), alternative: csv.alternative_txt, multiple:
toInt(csv.multiple), success: toInt(csv.success), suicide: toInt(csv.suicide), motive: csv.motive,
weapondetail: csv.weapdetail, nkill: csv.nkill, nkillter: csv.nkillter, nwound: csv.nwound, nwoundter:
csv.nwoundte, property: csv.property, propextent: csv.propextent, propextent_txt: csv.propextent_txt,
propvalue: csv.propvalue, propcomment: csv.propcomment, ransom: toInt(csv.ransom), ransomamt:
toInt(csv.ransomamt), ransomnote: csv.ransomnote, addnotes: csv.addnotes, dbsource: csv.dbsource});
```

```
//Eylemleri şehirler ile ilişkilendirme
```

```
using periodic commit load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';'
with toInt(csv.eylemid) as eylem, csv.sehir as sehir, toInt(csv.ulke) as ulke, toInt(csv.bolge) as bolge,
csv.eyalet as eyalet where sehir is not null and eyalet is not null match p = ((ci:Sehir {name: sehir})--
>(ps:Eyalet {name: eyalet})-->(co:Ulke {id: ulke})-->(r:Bolge {id: bolge})) with ci, eylem match
(e:Eylem {id: eylem}) create (e)-[:EYLEMIN_KONUMLANDIGI_YER]->(ci);
```

```
//Eylemleri saldırı tipleri ile ilişkilendirme
```

```
using periodic commit load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';'
with toInt(csv.eylemid) as eylem, toInt(csv.saldiritipi1) as saldiritipi where saldiritipi is not null match
(e:Eylem {id: eylem}), (at:SaldiriTipi {id: saldiritipi}) create (e)-[:SALDIRI_TURU]->(at);
```

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid)
as eylem, toInt(csv.saldiritipi2) as saldiritipi where saldiritipi is not null match (e:Eylem {id: eylem}),
(at:SaldiriTipi {id: saldiritipi}) create (e)-[:SALDIRI_TURU]->(at);
```

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, toInt(csv.saldiritipi3) as saldiritipi where saldiritipi is not null match (e:Eylem {id: eylem}), (at:SaldiriTipi {id: saldiritipi}) create (e)-[:SALDIRI_TURU]->(at);

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, toInt(csv.saldiritipi4) as saldiritipi where saldiritipi is not null match (e:Eylem {id: eylem}), (at:SaldiriTipi {id: saldiritipi}) create (e)-[:SALDIRI_TURU]->(at);

//eylemleri hedeflerle ilişkilendirme

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, csv.hedef1 as hedef where hedef is not null match (e:Eylem {id: eylem}), (t:Hedef {name: hedef}) create (e)-[:HEDEFLENEN]->(t);

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, csv.hedef2 as hedef where hedef is not null match (e:Eylem {id: eylem}), (t:Hedef {name: hedef}) create (e)-[:HEDEFLENEN]->(t);

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, csv.hedef3 as hedef where hedef is not null match (e:Eylem {id: eylem}), (t:Hedef {name: hedef}) create (e)-[:HEDEFLENEN]->(t);

//Grupları eylemler ile ilişkilendirme

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, csv.grupadi as grup where grup is not null match (e:Eylem {id: eylem}), (g:Grup {name: grup}) create (e)-[:GERCEKLESTIREN]-(g);

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, csv.grupadi2 as grup where grup is not null match (e:Eylem {id: eylem}), (g:Grup {name: grup}) create (e)-[:GERCEKLESTIREN]-(g);

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, csv.grupadi3 as grup where grup is not null match (e:Eylem {id: eylem}), (g:Grup {name: grup}) create (e)-[:GERCEKLESTIREN]-(g);

//Eylemleri silah tipleri ile ilişkilendirme

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, toInt(csv.silahtipi1) as silah where silah is not null match (e:Eylem {id: eylem}), (w:Silah {id: silah}) create (e)-[:KULLANILAN_SILAH]->(w);

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, toInt(csv.silahtipi2) as silah where silah is not null match (e:Eylem {id: eylem}), (w:Silah {id: silah}) create (e)-[:KULLANILAN_SILAH]->(w);

load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid) as eylem, toInt(csv.silahtipi3) as silah where silah is not null match (e:Eylem {id: eylem}), (w:Silah {id: silah}) create (e)-[:KULLANILAN_SILAH]->(w);

```
load csv with headers from "file:/gtdb_transformtr.csv " as csv fieldterminator ';' with toInt(csv.eylemid)
as eylem, toInt(csv.silahtipi4) as silah where silah is not null match (e:Eylem {id: eylem}), (w:Silah {id:
silah}) create (e)-[:KULLANILAN_SILAH]->(w);
```

```
match (n) return count(n); match ()-[r]->() return count(r);
```



ÖZGEÇMİŞ

Kutlu EREN, 1989 yılında Eskişehir’de doğdu. İlk ve orta öğrenimini Eskişehir’de tamamladı. 2007 yılında Pamukkale Üniversitesi Bilgisayar Mühendisliği bölümünü kazandı. 2009-2010 yıllarında Danimarka’da bulunan University of Southern Denmark’ta yazılım mühendisliği alanında eğitim aldı. 2011 yılında mezun olmasından itibaren İstanbul’da çeşitli şirketlerde yazılım ve teknoloji alanlarında çalıştı. 2013 yılında Sakarya Üniversitesi Yönetim Bilişim Sistemleri bölümünde yüksek lisansa başladı. 2015 yılında askerlik görevini tamamladıktan sonra Ankara ilinde Kale Yazılım bünyesinde Yazılım Uzmanı olarak kariyerine devam etmektedir.