

# 8

## Amaçlarımız

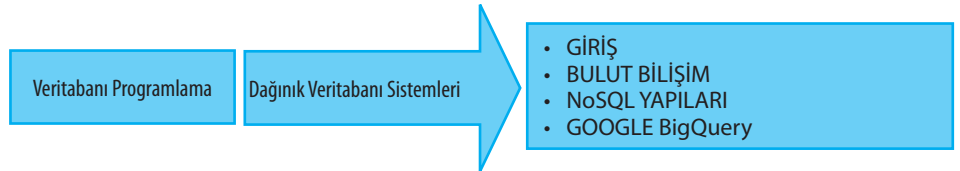
Bu üniteyi tamamladıktan sonra;

- Bulut bilişimi kavramını tanımlayabilecek,
- Bulut bilişim hizmet ve erişim modellerini açıklayabilecek,
- NoSQL veritabanı yapılarını tanımlayabilecek bilgi ve becerilere sahip olabileceksiniz.

## Anahtar Kavramlar

- Bulut Bilişim
- NoSQL
- Anaçatı Bilgisayar
- Google BigQuery
- Dağılık Veritabanı Sistemleri

## İçindekiler



# Dağınık Veritabanı Sistemleri

## GİRİŞ

Günümüzün oldukça moda konularından olan büyük veri kavramı, insanların ürettiği verinin artarak çoğaldığı ve artık verinin yönetilebilir boyutlardan çıktığı zaman ortaya çıkmış bir kavramdır. İnsanlar her gün sosyal medyada, blog sayfalarında, internet gezintilerinde milyarlarca satırlık veri oluşturmaktadır. Oluşan bu veri çöplüğünden anlamlı verilerin oluşturulması büyük veri kavramının ilgi alanına girer.

Verinin bu kadar arttığı bir ortamda onu işleyecek olan yapıların da ortaya çıkması kaçınılmazdır. Bu ihtiyaçtan dolayı verinin saklanması, işlenmesi, anlamlandırılması konusunda ucuz maliyetli çözümler sunulabilen bulut bilişim kavramı ortaya çıkmıştır. Bulut bilişim uzaktaki bir bilişim kaynağının son kullanıcıya ağ üzerinden uygun maliyet ile kullandırılmasıdır.

Bulut bilişim sayesinde elimizde oluşan bu büyük veriyi saklayabilecek kadar alan ve anlamlandırmak için işlem gücüne sahip olabildik ancak verileri saklamak ve aralarındaki ilişkileri tutmak için hâlihazırda geliştirilmiş olan ilişkisel veritabanı yapıları yetersiz kalmaktadır. Bu yetersizliği gidermek için NoSQL veritabanı yapıları geliştirilerek sorun çözülmeye çalışılmıştır.

## BULUT BİLİŞİM

**Bulut bilişim**, tek bir yerel sunucunun ya da kişisel bir cihazın kaynağına bağlı olarak yapılan hesaplama yerine birden fazla cihazın kaynaklarını beraber kullanarak hesaplama yapma işi olarak tanımlanabilir. Bulut bilişim isminde geçen “bulut” kelimesi internet ortamını kastetmektedir çünkü günümüzde uzaktaki bir bilişim kaynağını son kullanıcıya servis etmenin en kolay yolu internet üzerinden sunmaktır. Bu bağlamda bulut bilişime internet tabanlı hesaplama demek yanlış olmayacaktır. Bulut bilişim, gerek donanım kaynaklarını, gerek yazılım servislerini son kullanıcıya hizmet olarak sunabilen dinamik olarak ölçeklendirilebilen yapılardır. Bulut bilişim bu yönüyle bir ürün değil, hizmettir. Bulut bilişim sayesinde internet üzerinden veritabanı, süper hesaplama, dosya paylaşımı vb. gibi hizmetler son kullanıcıya kolaylıkla sunulabilir. Kavramsal olarak bulut bilişimin temelleri 1950'lere dayanmaktadır. Şirketler için **anaçatı bilgisayar** yatırımları oldukça maliyetli yatırımlardır. Bu sebepten dolayı şirketler, anaçatı bilgisayardan en iyi şekilde yararlanmak için işlem gücü olmayan zayıf terminalleri kullanmışlardır. Bu zayıf terminaller zaman paylaşımlı olarak birden fazla kullanıcının anaçatı bilgisayar yapılarına erişmesini ve işlemci gücünü kullanmasını sağlıyordu. Böylelikle anaçatı bilgisayarın işlemcilerinin boş kalma süreleri en aza indirilerek bu pahalı bilgisayarlardan en iyi şekilde yararlanılması

**Bulut bilişim** İngilizce “cloud computing” terimine Türkçe’de verilen isimdir.

**Anaçatı bilgisayar**, (mainframe) yüzlerce kullanıcıya aynı zamanda farklı hizmetler verebilen içerisinde çok sayıda işlemci ve sabit disk barındıran güçlü ve pahalı bilgisayarlardır.

sağlanmıştır. Günümüzde kullanılan anlamı ile bulut bilişim fikri Leonard Kleinrock tarafından “bilgisayar ağları henüz emekleme döneminde, fakat bir süre sonra daha karışık bir hâle gelecek ve muhtemelen birtakım bilgisayar servislerini, nasıl günümüzde kullanılan elektrik telefon servisleri gibi evlerimizde ve tüm ülke çapında kullanabiliyor olacağız” öngörüsü ile 1969 yılında atılmıştır.

## Bulut Bilişim Erişim Modelleri

Bulut bilişim sistemleri erişim modellerine göre dörde ayrılırlar. Bu sınıflandırma genel bulut, topluluk bulut, özel bulut ve karma bulut şeklindedir.

Genel bulut yapılarında depolama, yazılım ve diğer kaynaklar hizmet sağlayan şirket tarafından genel son kullanıcılara sunulur. Sunulan hizmet kullandığın kadar öde modeli ile ya da ücretsiz olarak kullanıcılara servis edilir. Sunuculara erişim internet üzerinden sağlanır. Amazon EC2, Google AppEngine, Dropbox vb. ürünleri genel bulut yapısı içerisinde sayabiliriz.

Topluluk bulut hizmeti ortak ihtiyaçları olan ve bu ihtiyaçları beraber karşılama amaçları olan çeşitli organizasyonların altyapılarını birbirleri ile paylaşarak kaynakları daha etkin bir biçimde kullandıkları bulut yapılarıdır.

Özel bulut yapıları sadece tek bir organizasyon için işletilen yapılarıdır. Bu tür bulutların sadece şirket içi kullanımına izin verilmektedir, dış dünya/son kullanıcı ile herhangi bir bağlantıları yoktur. Bu tür bulut yapıları içerisinde şirkete ait her türlü hassas veri, yazılım kaynağı sadece şirket içerisinde güvenle paylaşılabilir.

Karma bulut yapıları en az iki veya daha fazla özel, topluluk veya genel bulut yapılarının birleşerek oluşturduğu yapılarıdır. Karma bulut yapıları içerisinde hem kurumlara ait hassas bilgiler güvenli bir şekilde paylaşılabilirken hem de bazı hizmetler dışarıdaki son kullanıcı ile de paylaşılabilir.

## Bulut Bilişim Hizmet Modelleri

Bulut bilişim yapılarında her şey kullanıcıya hizmet olarak sunulur. Servisler belirli işleri yapması için otomatikleştirilmiş alt yordamların toplamı olarak tarif edilebilir. Bu tanımdan yola çıkarak servislerin hizmet olarak sunulmasını üç ana başlık altında toplayabiliriz. Bu başlıklar hizmet olarak altyapı, hizmet olarak platform, hizmet olarak yazılım şeklinde sıralanır.

**IaaS** İngilizce “Infrastructure as a Service” kelimelerinin ilk harflerinin birleşmesi ile oluşmuştur.

Hizmet olarak altyapı (**IaaS**), teknoloji işletmeleri veya geliştiriciler için dosya saklama alanı ya da hesap yapabilme gibi kaynakları hizmet olarak sunulduğu yapılarıdır. Bunların yanı sıra işletim sistemlerinin sanallaştırılarak son kullanıcıya sunulması teknolojisi de bu kapsamda yer almaktadır. Kullanıcılar nasıl elektrik su gibi hizmetler için kullandığı kadar ödeme yapabiliyorsa aynı durum burada da geçerlidir. Kullanıcılar pahalı yatırımlar yerine başka bir şirketin yaptığı altyapı ve yazılım yatırımını kullanacakları kadar kiralayabilirler. Ayrıca bu yapılar anlık ortaya çıkan talep artışları ya da talep azalışları karşısında kiralama hizmeti dinamik olarak ölçekli yükseltilebilir ya da ölçekli küçültülebilir. Hizmet olarak altyapı ürünlerine örnek verecek olursak Amazon EC2, Amazon Web Services, BlueLock, PrimaCloud gibi uygulamaları sayabiliriz.

INTERNET



**Amazon EC2 hizmeti hakkında daha fazla bilgi edinmek için <https://aws.amazon.com/ec2/> adresini ziyaret edebilirsiniz.**

**PaaS** İngilizce “Platform as a Service” kelimelerinin ilk harflerinin birleşmesi ile oluşmuştur.

Hizmet olarak platform (**PaaS**), teknoloji işletmeleri ve geliştiriciler için bulut bilişime hazır çözümler üretmesini sağlayan yapıların sunulduğu modeldir. Son kullanıcı belirli altyapılar üzerine kurulmuş olan bu yazılım platformunu kullanarak yazılım geliştirebilir

ya da geliştirdiği yazılımı çalıştırabilir. Platform hizmetinin geliştiricilere en temel faydası yazılım geliştirme ve yazılım dağıtım süreçlerini tek elden yönetebilir olmasıdır. Hizmet olarak platform ortamları ortaya çıkabilecek olan ihtiyaca göre dinamik olarak ölçekli yükseltilebilir ya da ölçekli olarak küçültülebilir. Hizmet olarak platform ürünlerine örnek verecek olursak ActiveState Stackato, Apprenda, Amazon Elastic Beanstalk, Microsoft Azure, Centurylink Appfog, CloudControl dotCloud, Engine Yard, Google App Engine, IBM Bluemix, Pivotal Cloud Foundry, Red Hat OpenShift, Heroku gibi ürünleri sayabiliriz.

**Google App Engine hizmeti hakkında daha fazla bilgi edinmek için <http://appengine.google.com> adresini ziyaret edebilirsiniz.**



İNTERNET

Hizmet olarak yazılım (**SaaS**), belirli bir amaç için geliştirilmiş yazılımları son kullanıcıya kiralama yöntemi ile sunmamızı sağlayan bulut yapılarıdır. Yazılım sağlama hizmeti ile kullanıcıların ihtiyaç duyduğu CRM, ERP, finans ve muhasebe, ofis uygulamaları, e-posta yazılımları gibi programları bulut üzerinde dağıtılır. Bu yöntem ile son kullanıcı yazılımın geliştirilmesi, çalıştırılması gibi maliyetler yazılımı kullanan müşteriler arasında paylaşılabileceğinden bu maliyetler en aza inmiş olur. Genellikle bu modelde yazılıma web tarayıcı üzerinden erişim sağlanır, bu durum son kullanıcıya herhangi ek bir yazılım kurmadan tüm dünya üzerinden yazılımı kullanabilmesi anlamına gelmektedir. Bu modelde de diğer hizmet modellerinde olduğu gibi, kullanıcılar aldıkları hizmet kadar ücretlendirilirler. Bu modeli kullanan yazılımlara örnek verecek olursak Salesforce, Workday, NetSuite, ServiceNow, Athenahealth, Microsoft Office Online, Google Apps, Concur, Citrix GoToMeeting, Cisco WebEx gibi yazılımları sıralayabiliriz. Ayrıca Türkiye'nin en büyük internet sağlayıcısı olan TTNET; TTNET Bulutu adlı hizmeti ile Türkiye piyasasındaki yerini almıştır.

**SaaS** İngilizce "Software as a Service" kelimelerinin ilk harflerinin birleşmesi ile oluşmuştur.

## NoSQL YAPILARI

Son on beş yılda internetin kullanımının artması ve twitter, facebook gibi sosyal medyanın hayatımıza girmesi ile insanların sanal dünya için oluşturduğu verinin, her geçen gün artan bir ivme ile büyümesine sebep olmuştur. **NoSQL** kavramı ilk defa 1998 yılında ortaya atılmıştır ve zaman içerisinde gelişerek günümüze kadar gelmiştir. Bir diğer deyişle NoSQL kavramı, internete yüklenen ve gün geçtikçe artan bu veriyi depolayabilmek, yüksek sayıdaki kullanıcıların anlık taleplerine cevap vermek amacı ile ortaya çıkmış yatay olarak ölçeklendirilebilen veritabanı sistemlerine verilen genel isimdir.

Günümüzde, oluşan bu kadar büyük verinin analizinin ve sorgulamalarının yapılması için gerekli olan elemanlar NoSQL kavramı içerisinde bulunmaktadır. İlişkisel veritabanları kararlı çalışması ve veri bütünlüğünün korunması için **ACID** bölünmezlik, tutarlılık, izolasyon, dayanıklılık kurallarını içermektedir. Ancak NoSQL yapılarında **hareket** kavramı bulunmadığından ilişkisel veri tabanlarının uyduğu kurallara uymayabilirler.

Eric Brewer tarafından 1998 yılında ortaya konan **CAP** teoremi ya da bir diğer adı ile Brewer teoremi dağıtık sistemlerin aynı anda tutarlılık, ulaşılabilirlik ve bölünebilirlik toleransı koşullarına sahip olamayacağını söyler. Teoreme göre yukarıda bulunanlardan sadece ikisi aynı anda bir dağıtık sistemde bulunabilir. Yani dağıtık bir sistemin her alt bileşeninden aynı veriye erişebilme, aynı anda bütün isteklere cevap verebilme, kaybedilen paket olsa bile verinin bütününe kaybetmeme özelliklerine aynı anda sahip olması mümkün değildir. Yukarıda sayılan üç özellikten birinde mutlaka problem çıkacaktır.

2015 yılında Twitter kullanılarak günlük yaklaşık 500 milyon tweet atılmaktadır.

**NoSQL** için "not only sql" tabiri yani "sadece sql değil" de kullanılmaktadır.

**ACID** kelimesi Atomicity, Consistency, Isolation, Durability kelimelerinin ilk harflerinin birleşmesi ile oluşmuştur.

**Hareket** veritabanında yapılan işlemler olarak adlandırılır. İngilizce "transaction" teriminin Türkçedeki karşılığıdır.

**CAP** kelimesi İngilizce "Consistency", "Availability" ve "Partition Tolerance" kelimelerinin ilk harflerinden oluşturulmuştur. Türkçede sırası ile tutarlılık, ulaşılabilirlik ve bölünebilirlik toleransı kelimelerine karşılık gelmektedir.

Yukarıda anlatılan iki teoremden yola çıkarak, CAP teoremi dağıtık bir sistemin sahip olması gereken özellikleri, ACID teoremi verinin ve hareket kavramının nasıl yapılacağına dair kuralları ortaya konulmuştur. Günümüzde NoSQL sistemler de bu kavramlar göz önünde tutularak oluşturulmuştur. Dilerseniz NoSQL kavramının özelliklerine bakalım.

- Şema Bağımsız: Bildiğiniz üzere ilişkisel veritabanı yapılarında, veritabanı şeması veritabanının saklayabileceği tüm veri ve veri yapılarının tanımını göstermektedir. NoSQL veri tabanlarında tanımlı bir şema bulunmamaktadır.
- İlişkisel veritabanında tablolar arasındaki bağlantı, tabloların birbirleri ile ilişkili olan alanları üzerinden kurulur. NoSQL veri tabanlarındaki kayıtlar bütün ilişkiyi içeren bir satır olarak tutulur.
- Büyük boyutlardaki ilişkisel veritabanı yapılarını çalışmak için yüksek maliyetli sunuculara ihtiyaç duyarken, NoSQL veritabanı standart sunucu ile çalışabilir. Daha yüksek hacimli verinin tutulabilmesi için standart sunuculardan daha fazla kullanmak gereklidir.
- NoSQL veritabanı yapıları kolaylıkla dağıtık bir biçimde çalıştırılabilirler.
- NoSQL veritabanı yapıları ilişkisel veritabanlarına göre daha büyük veriyi içerisinde saklayabilir.

Yukarıda NoSQL yapılarının ilişkisel veritabanlarına göre avantajlarından bahsettik. Ancak bazı önemli konularda ilişkisel veritabanının NoSQL yapılar üzerinde üstünlüğü bulunmaktadır.

- İlişkisel veritabanı yapılarının kararlı çalışma ve veri bütünlüğünün sağlanması ACID kuralları çerçevesinde yapılmaktadır.
- Verilerin okunup yazılması bir veritabanı şemasına bağlıdır.
- İlişkisel veritabanı yapıları gerçek zamanlı sorgu yönetimini içerir.
- Tablolardaki alanlar üzerinden **birleştirme** ve **gruplama** gibi karmaşık sorgular çalıştırılabilir.

İlişkisel veritabanı yapılarında tablolar üzerinde **birleştirme** "join" komutu ile yapılabilir.

İlişkisel veritabanı yapılarında tablolar üzerinde **gruplama** "group by" komutu ile yapılır.

## NoSQL Veritabanı Tipleri

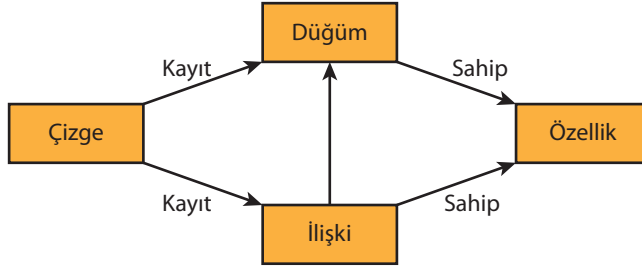
Günümüzde NoSQL veritabanı yapıları için kabul görmüş bir standart bulunmamaktadır. Standardın oluşmamasından dolayı çeşitli şirketler birbirlerinden farklı sorunları çözmeye odaklanmıştır. Bu farklı sorunlara odaklanmanın sonucu olarak birbirlerinden farklı çalışan, farklı özellikler barındıran birçok NoSQL tipinde veritabanı yapısı ortaya çıkmıştır. Günümüzde 200 den fazla NoSQL veritabanı ürünü vardır ve her geçen gün bu sayı artmaktadır. Genel olarak NoSQL veritabanılarını gruplayacak olursak dört farklı kategoriye ayırabiliriz.

### Çizge Tabanlı NoSQL Yapıları

Bu tip NoSQL veritabanı türünde veriler, veritabanı içerisinde çizge şeklinde tutulur. Birbirleri ile ilişkili olan verilerin tutulması gerektiğinde bu tür NoSQL yapısı kullanım için oldukça uygundur. İlişkiye örnek verecek olursak Facebook üzerinde birçok kişi birbirini takip ediyor, dahası birbirleri ile ilgili resim, video, yorum vb. birçok nesne birbiri ile ilişkilendiriliyor. Bu tür durumlarda çizge tabanlı yapılar oldukça iş görmektedir. Facebook tarafından geliştirilen "Facebook Graph Search" ürünü ile çizge üzerinden "Ali'nin Eskişehir'de Ahmet ile etiketlendiği fotoğrafları" gibi sorguları yapmak mümkündür. Veriler ve ilişkileri Şekil 8.1'de olduğu gibi tutulur.

Şekil 8.1

Çizge Tabanlı NoSQL Yapıları



**Kaynak:** <https://www.linkedin.com/pulse/choosing-right-NoSQL-database-type-app-development-snehesh-mitra>

Çizge tabanlı veritabanlarındaki arama motorları çizge üzerinde dolaşmak için gerekli olan algoritmaları içerir. Örneğin aşağıdaki örnek kod parçasında Neo4j NoSQL veritabanı üzerinde “*Breadth First Search*” algoritması kullanılarak Ahmet’in arkadaşlarını bulan ve ekranda listeleyen kod parçası listelenmiştir.

```
// Ahmet'in arkadaşlarını döndürür.
Traverser friendsTraverser = Ahmet.traverse(
    Traverser.Order.BREADTH_FIRST,
    StopEvaluator.END_OF_NETWORK,
    ReturnableEvaluator.ALL_BUT_START_NODE,
    RelTypes.KNOWS,
    Direction.OUTGOING );

// Ahmet'in arkadaşlarını ekranda listeler.
System.out.println( "Ahmet'in Arkadaşları:" );
for ( Node friend : friendsTraverser )
{
    System.out.printf( "At depth %d => %s\n",
        friendsTraverser.currentPosition().getDepth(),
        friend.getProperty( "name" ) );
}
```

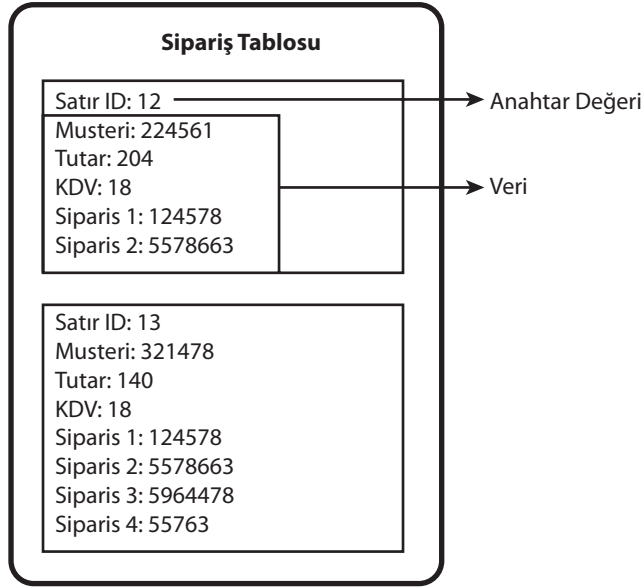
Yukarıda da anlatıldığı gibi çizge tabanlı veritabanı yapıları, en iyi sonuçları verilerin birbirleri ile ilişkili olarak tutulması gerektiği zamanlarda verir. Örneğin sosyal ağlar, öneri motoru uygulamaları, mekânsal veri ve haritalama gibi konularda oldukça iyi sonuçlar vermektedir. Ayrıca geleceğe yönelik tahmin, öneri ve sonuç analiz motorlarında da kullanılması oldukça iyi sonuçlar vermiştir. Çizge tabanlı veritabanı ürünlerine örnek verecek olursak Neo4J, ArangoDB, FlockDB, Infinite Graph, Sparksee, TITAN, InfoGrid, HyperGraphDB, Trinity gibi ürünleri sıralayabiliriz.

### Anahtar-Değer Tabanlı NoSQL Yapıları

Anahtar-Değer tabanlı olan yapılar, NoSQL veritabanlarında kullanılan en temel yöntemlerden biridir. Veri, anahtar ve buna karşı gelen değer şeklinde kaydedilir ve sorgulamalar anahtar üzerinden yapılır. Anahtar-Değer veritabanı yapılarının en zayıf oldukları taraf, anahtarlar üzerinde yapılacak arama işlemlerinin yüksek maliyetli olmasıdır. Bu zayıflığın aşılması için Google tarafından geliştirilen BigTable aracı gibi araçlarla anahtarların sıralanması ve kümelenmesi sağlanarak bu zayıflık aşılmıştır. Anahtar-değer veritabanı yapılarında veriler Şekil 8.2’de olduğu gibi tutulur.

Şekil 8.2

Anahtar-Değer  
Tabanlı NoSQL  
Yapıları



**Kaynak:** <http://bi-bigdata.com/2012/12/25/what-is-key-value-stores/>

Anahtar-Değer tabanlı NoSQL yapıları, kullanıcıların oturum bilgilerinin, şema bağımsız kullanıcı veya sipariş gibi bilgilerin tutulmasında oldukça etkin olarak kullanılabilirler ancak veriler arasında ilişkiler kurulacaksa, veriler üzerinde sorgulama yapılacaksa bu tür yapıları kullanmak çok uygun değildir. Bu tür yapılara örnek olarak Redis, Riak, Amazon DynamoDB verilebilir.

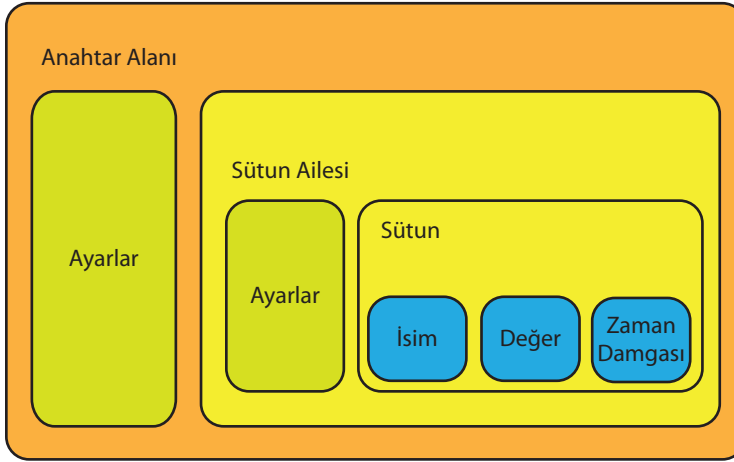
### Sütun Tabanlı NoSQL Yapıları

Bu tür veritabanı yapıları çok miktarda veri tutmak için kullanılır. Anahtar-Değer veritabanlarında olduğu gibi burada da anahtar mevcuttur ancak anahtar tek bir kayıt yerine sütunu gösterir. Bir başka deyişle sütun tabanlı yapılarda veriler satır olarak değil sütunlarda gruplanmış olan hücreler şeklinde kaydedilirler. Yapı içerisindeki sütun aileleri sanal olarak oluşturulmuş sonsuz sayıda kolonu içlerinde barındırabilirler. Okuma ve yazma işlemleri tanımlanan sütunlar üzerinden yapılır. Bu tür yapılardaki veri modeli aşağıdaki gibi tanımlanabilir.

- Sütun Ailesi: Sütun ve süper sütunları içerisinde kolaylıkla gruplayabilen yapıdır.
- Anahtar: Kayda ait değişmeyen adıdır.
- Anahtar Alanı: Bu yapının en üst katmanını belirler, örneğin uygulamanın adı yani veritabanının adı olabilir.
- Sütun: Verilerin isim ve değer ile tutulduğu sıralı listelerdir.
- Küme: Birden fazla fiziksel sunucuyu ya da birden fazla anahtar alanını içerisinde barındıran yapılardır.

Sütun tabanlı NoSQL veritabanlarında verinin nasıl tutulduğunu daha iyi anlamak için Şekil 8.3'te görünen Cassandra isimli ürünün veri modeline bakabilirsiniz.

Şekil 8.3

Cassandra NoSQL  
Veritabanı Veri  
Modeli

**Kaynak:** <http://www.slideshare.net/sadeghsalehi2/NoSQL-database-cassandra-column-base-db>

Bu tür NoSQL veritabanı yapılarının kullanım alanlarına bakacak olursak; içerik yönetim sistemleri, blog platformları, sosyal medya, işlem kaydı analiz motorları gibi yapılarda sıklıkla kullanılırlar. Bu tür yapıları kullanan ürünlere örnek verecek olursak Cassandra, Apache Hadoop HBase gibi yazılımları sıralayabiliriz.

### Belge Tabanlı NoSQL Yapıları

Temel olarak ilgilendiği alan belgelerin yönetilmesidir. Çalışma prensibi olarak sütun temelli yapılar gibi çalışır ancak iç içe geçmiş belgeler arasında ilişki kurmak bu yapılar ile daha kolaydır. Bu tür veritabanlarına en iyi verilecek örneklerden biri MongoDB isimli üründür. MongoDB aşağıdaki kavramları kullanarak verilerini saklar.

- Veritabanı: MongoDB veritabanı, koleksiyonların fiziksel olarak barındırıldığı yerdir.
- Koleksiyon: MongoDB belgelerinin kayıtlı olduğu gruplardır, Geleneksel ilişkisel veritabanındaki karşılığı tablodur.
- Belge: Belgeler, geleneksel veritabanlarındaki kayıt (sıra) kavramıyla aynıdır. Her belge yani sıra, koleksiyonlar içerisinde XML, JSON, BINARY JSON (BSON) gibi formatında saklanır.

Bu tür veritabanları içerik yönetim sistemleri, blog platformları, analiz platformları, e-ticaret platformları gibi yapılarda kullanmak için oldukça kullanışlıdır. Konuyu daha iyi anlayabilmemiz için aşağıdaki MongoDB belge ekleme betiği örneğini inceleyebiliriz. Bu örnekte iki adet belge listelenmiştir. İçeriğinde “MongoDB Kaydı” olan kayıta herhangi bir yorum satırı yoktur ancak ‘NoSQL Veritabanı’ olan kayıta bir kullanıcının yorum kaydı da listelenmiştir.

```

>db.post.insert([
{
  Baslik: 'MongoDB Kaydı',
  Tanim: 'MongoDB bir NoSQL veritabanıdır',
  Kaydeden: 'Uğur GÜREL',
  Anahtar_Kelime: ['mongodb', 'database', 'NoSQL'],
  OkunmaSayisi: 100
},
{
  Baslik: 'NoSQL Veritabanı',
  Tanim: 'NoSQL Geleceğin Teknolojisidir',

```



```

Kaydeden: 'Tuna GÜREL',
Anahtar_Kelime: ['mongodb', 'database', 'NoSQL'],
OkunmaSayisi: 20,
Yorum: [
  {
    Kisi: 'internet',
    Yorumİcerik: 'Çok yararlı bir yayın',
    Tarih: new Date(2015,11,2,2,35),
    Begeni: 0
  }
]
})

```

SIRA SİZDE



Sizde öğrencilerin adının soyadının sınıfının ve öğrenci ile ilgili yorumların tutulduğu bir MongoDB belge ekleme betiği kaydı oluşturunuz içerisinde iki adet örnek öğrenci kaydı yerleştiriniz.

**Büyük veri** İngilizce "big data" terimine Türkçe'de verilen isimdir.

## GOOGLE BigQuery

Google BigQuery, Google şirketinin 2012 yılında başlattığı **büyük veri** analiz hizmeti sunduğu yapıdır. Son kullanıcılar Google BigQuery yapısını kullanarak ücretsiz ya da kullandığın kadar öde modeli ile hizmet alabilmektedirler. Google BigQuery'nin bu kadar büyük veriler içerisinde bu kadar hızlı sorgulara cevap vermesinin altında yatan teknoloji yine Google tarafından geliştirilmiş olan Dremel altyapısıdır.

İNTERNET



Google BigQuery fiyatlandırma politikasını hakkında daha fazla bilgi edinmek için <https://cloud.google.com/bigquery/pricing> adresini ziyaret edebilirsiniz.

**SQL**, Yapılandırılmış Sorgu Dili (Structured Query Language) veritabanı üzerinde işlemler yapmamızı sağlayan komutları içerisinde barındıran programlama dilidir.

Google BigQuery'ye web arayüzü üzerinden, komut satırı üzerinden ya da BigQuery REST API arayüzlerinden birini kullanarak ulaşabilirsiniz. Dremel, MapReduce'e nazaran, veri kümesi sorgu işlemlerini daha kısa zamanda gerçekleştirme yeteneğine sahiptir. Ayrıca Dremel alışıla gelmiş **SQL** dilinde bir sorgulama formatı kullanır.

## Google BigQuery Servisi

Google BigQuery yapısı üzerinde projeler, veri bloğu, tablolar ve işler olmak üzere dört temel bileşen vardır. Konunun daha iyi anlaşılabilmesi için bu dört temel bileşen hakkında bilgi verilecektir.

### Projeler

Projeler, Google BigQuery yapısı üzerindeki en üst düzeydeki bileşendir. Diğer iki bileşen proje içerisinde yer almaktadır. Projelere erişim hakları, erişim düzeyleri tanımlanabilir.

### Veri Bloğu

Veri blokları içerisinde bir ya da daha fazla tabloyu barındıran yapılardır. İşlem yapabilmek için proje içerisinde en az bir tane veri bloğu oluşturulmuş olması gereklidir.

### Tablolar

Tablolar kullanıcıya ait verilerin tutulduğu yapılardır. Her tablonun kendine ait bir şeması olabilir. Google BigQuery hem görünüm hem de bulut depoda tutulan dış tabloları desteklemektedir.

## İşler

İşler sizin tanımladığınız ve sizin adınıza veri yükleme, veriyi dışarı aktarma, veri üzerinde analiz yapma gibi işlemleri asenkron olarak çalıştıran yapılardır.

**Google BigQuery servisini kullanabilmek için Google hesabınızın olması gereklidir.**

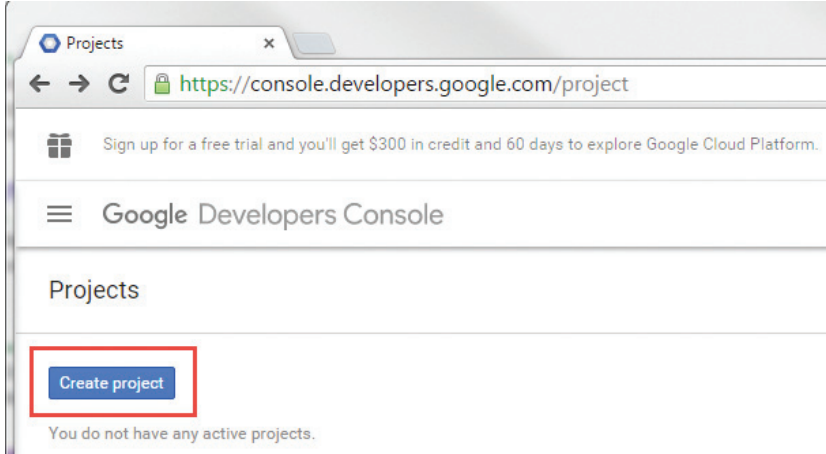


DİKKAT

İlk önce <https://console.developers.google.com> adresine bağlanıp Resim 8.1'de görünen "Create project" düğmesine basarak yeni bir proje oluşturmalıyız.

**Resim 8.1**

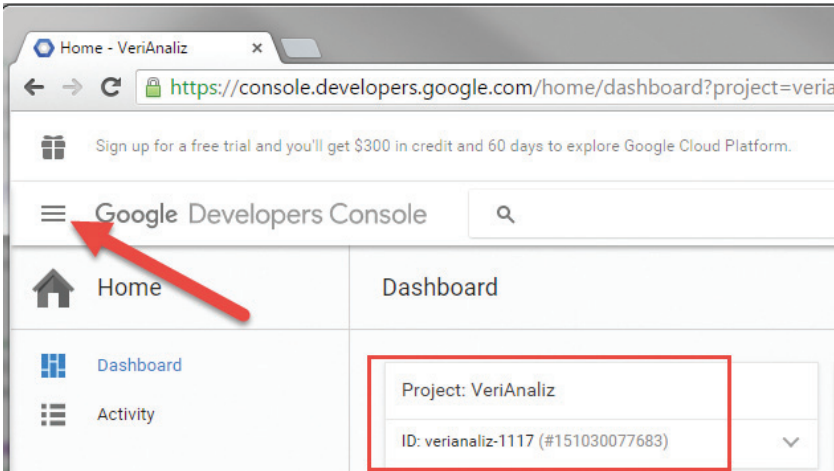
*VeriAnaliz  
İsimli Projenin  
Oluşturulması.*



Resim 8.1'de görünen düğmeye bastığımızda bizden bir proje ismi girmemizi isteyen bir ekran ortaya çıkar. Örnek olarak projenin adını "VeriAnaliz" olarak isimlendirelim. Resim 8.2'de görüldüğü gibi projemiz oluşturuldu.

**Resim 8.2**

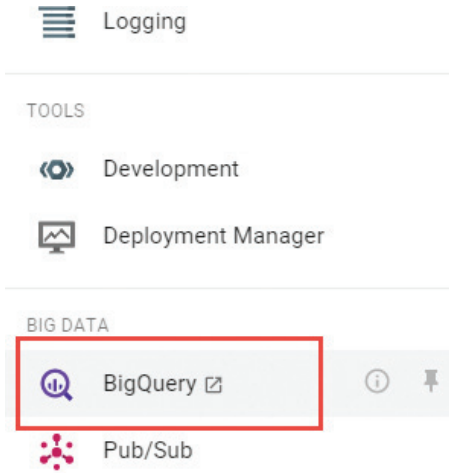
*VeriAnaliz  
İsimli Projenin  
Oluşturulması.*



Projemizi oluşturduktan sonra Resim 8.2'de ok ile gösterilen düğmeye basarak projemizin detaylarını açıyoruz, Google BigQuery ekranına ulaşmak için Resim 8.3'te görünen "BigQuery" düğmesine basıyoruz.

**Resim 8.3**

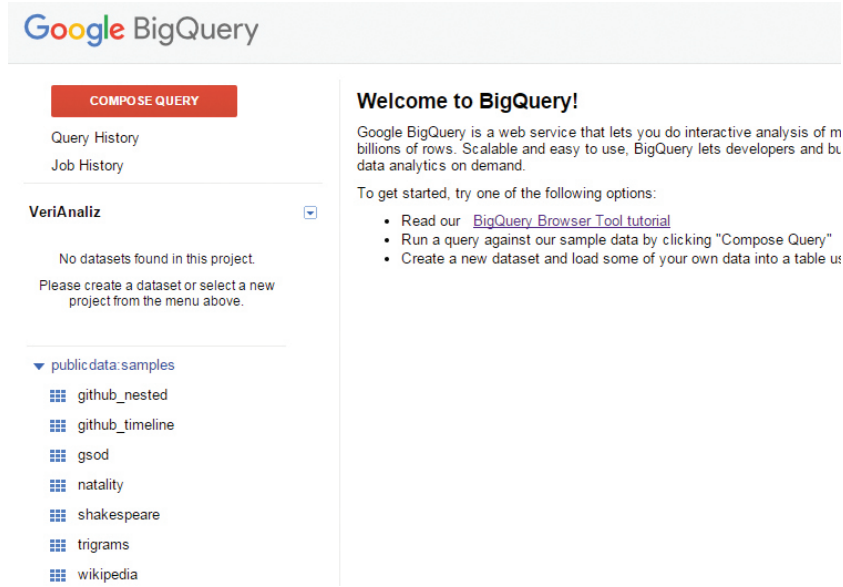
Google BigQuery  
Ekranına Ulaşmak.



Bizi Resim 8.4'te görüldüğü gibi bir hoş geldin ekranı karşılamaktadır. Bu ekrandan sorgu yapabilir, veri yükleyebilir ya da hâlihazırda var olan örnek verileri görebiliriz.

**Resim 8.4**

Google BigQuery Hoş  
Geldin Ekranı.



Proje oluşturulduğunda üzerinde sorgu da yapabileceğimiz örnek veri bloğuda yer alır. Örnek veri bloğu olan “Publicdata:samples” içerisinde yedi adet örnek tablo bulunmaktadır. Aşağıda bu örnek tabloların tuttuğu veriler ile ilgili detaylar listelenmiştir.

- i. Gsod isimli tablo 1929’ların sonlarından 2010’un başlarına kadar olan yağış miktarı, hava durumu, sıcaklık ve daha birçok hava verisini içeren tablodur. Veriler NOAA tarafından Amerika Birleşik Devletleri toprakları için toplanmıştır. Tablonun büyüklüğü yaklaşık 16 GB’dır ve içerisinde 114.420.316 adet satır bulunmaktadır.

- ii. Github\_nested isimli tablo GitHub yazılım deposu üzerinde yapılan proje indirme ve yorum ekleme ve birçok bilgiyi iç içe geçmiş şema şeklinde tutar. Tablonun büyüklüğü 1.58 GB'dır ve içerisinde 2.541.639 adet satır bulunmaktadır.
- iii. Github\_timeline isimli tablo GitHub yazılım deposu üzerinde yapılan proje indirme ve yorum ekleme ve birçok bilgiyi düz şema şeklinde tutar. Tablonun büyüklüğü 3.54 GB'dır ve içerisinde 6.219.749 adet satır bulunmaktadır.
- iv. Natalty isimli tablo Amerika Birleşik Devletlerinde 1969'dan 2008'e kadar olan sürede 50 eyalette doğan çocukların verilerini tutar. Tablonun büyüklüğü 21.9 GB'dır ve içerisinde 137.826.763 satır veri barındırır.
- v. Shakespeare isimli tablo Shakespeare'e ait eserlerde bulunan bir sözcük dizininin hangi eserde kaç kere tekrar ettiğinin sayısını tutar. Tablonun büyüklüğü 6.13 MB'dır ve içerisinde 164.656 satır veri barındırır.
- vi. Trigams isimli tablo 1520 ile 2008 arasında yayınlanmış olan örnek eserlerdeki üçlü kelime öbeklerini (örneğin "through projects and") ve sayılarını tutar. Tablonun büyüklüğü 258 GB'dır ve içerisinde 68.051.509 satır veri barındırır.
- vii. Wikipedia isimli tablo Wikipedia sitesindeki verilerin 2010'a kadar olan tüm revizyonlarını tutar. Tablonun büyüklüğü 35.7 GB'dır ve içerisinde 313.797.035 satır veri barındırır.

Eğer herhangi bir tablonun şemasına ulaşmak istersek ilgili tablonun ismine tıklamamız yeterlidir. İlgili tablonun ismine tıkladığımızda açılan ilk ekran veritabanı şemasının gösterildiği "schema" isimli ekrandır. Resim 8.5'te de görüldüğü üzere ekranda sütunların isimleri ve veri tipleri görünür, ayrıca kolonun tanımını da bu ekran üzerinden ekleyebiliriz.

Resim 8.5

Tablo Şema Ekranı.

## Table Details: github\_nested

Schema

Details

Query Table

## Schema

repository	RECORD	NULLABLE	Describe this field...
repository.url	STRING	NULLABLE	Describe this field...
repository.has_downloads	BOOLEAN	NULLABLE	Describe this field...
repository.created_at	STRING	NULLABLE	Describe this field...
repository.has_issues	BOOLEAN	NULLABLE	Describe this field...
repository.description	STRING	NULLABLE	Describe this field...
repository.forks	INTEGER	NULLABLE	Describe this field...
repository.fork	BOOLEAN	NULLABLE	Describe this field...
repository.has_wiki	BOOLEAN	NULLABLE	Describe this field...
repository.homepage	STRING	NULLABLE	Describe this field...

Detay (Details) düğmesine tıkladığımızda ise Resim 8.6'da görünen tablo ile ilgili detay bilgiye ulaşabiliriz.

Resim 8.6

Detay Ekranı.

## Table Details: github\_nested

Schema

Details

Query Table

## Description

Describe this table...

## Table Info

Table ID	publicdata:samples.github_nested
Table Size	1.58 GB
Number of Rows	2,541,639
Creation Time	Sep 28, 2012, 12:49:47 AM
Last Modified	Sep 28, 2012, 12:49:47 AM
Data Location	US

Son olarak bizi sorgu ekranına götürecek olan “Query Table” düğmesine bastığımızda Resim 8.7’de görünen sorgu ekranından ilgili tablo üzerinde standart sql cümleleri kullanarak sorgu yapabiliriz.

Resim 8.7

Sorgu Ekranı.

New Query ?

Query Editor

UDF Editor

X

```
1 SELECT * FROM [publicdata:samples.github_nested] LIMIT 1000
```

RUN QUERY

Save Query

Save View

Format Query

Show Options

✓

Bu ünite de örneklerimizi “publicdata:samples” veri bloğu üzerinden yapacağız. Sorgu yapabilmek için herhangi yeni bir komut öğrenmenize gerek yoktur. Sorgularınızı standart sql komutlarını kullanarak yapabilirsiniz.

İNTERNET



Google BigQuery tarafından desteklenen sql komutlarının tümüne <https://cloud.google.com/bigquery/query-reference> adresinden ulaşabilirsiniz.

Dilerseniz örnek olarak “samples.gsod” isimli tablodan 116970 nolu istasyonun (station\_number) 1989 yılından sonraki yıl (year), ay (month), gün (day) ve ortalama sıcaklık (mean\_temp) verilerini süzelim ve veriyi de yıl, ay ve güne göre sıralayalım.

Bu sorgu için kullanacağımız sql cümlesi aşağıdadır. Sorguyu çalıştırmak için Resim 8.7’de görünen ekrandaki “RUN QUERY” düğmesine basmamız yeterlidir.

```
SELECT
  year, month, day,
  mean_temp
FROM
  [publicdata:samples.gsod]
WHERE
```

```

station_number=116790
AND year > 1989
ORDER BY
year, month, day

```

Sorguyu Google BigQuery ekranında çalıştırdığımızda 4.26 GB verinin analiz edilerek 2.2 saniye içerisinde 7405 satırın listelendiğini sorgu sonucu ekranında (Resim 8.8) görebilirsiniz.

Resim 8.8

Sorgu Sonucu.

The screenshot shows the Google BigQuery 'New Query' interface. The query editor contains the following SQL code:

```

1 SELECT
2   year,
3   month,
4   day,
5   mean_temp
6 FROM
7   [publicdata:samples.gsod]
8 WHERE
9   station_number=116790
10  AND year > 1989
11 ORDER BY

```

Below the query editor, there are buttons for 'RUN QUERY', 'Save Query', 'Save View', 'Format Query', and 'Show Options'. A status message indicates 'Query complete (2.2s elapsed, 4.26 GB processed)' with a green checkmark.

The 'Query Results' section shows the results for 'Nov 7, 2015, 8:31:25 AM'. There are buttons for 'Download as CSV', 'Download as JSON', and 'Save as Table'. The results are displayed in a table with columns 'Row', 'year', 'month', 'day', and 'mean\_temp'.

Row	year	month	day	mean_temp
1	1990	1	1	27.700000762939453
2	1990	1	2	27.5
3	1990	1	3	26.100000381469727
4	1990	1	4	25.899999618530273
5	1990	1	5	23.700000762939453

At the bottom, there are navigation links: 'First', '< Prev', 'Rows 1 - 5 of 7405', 'Next >', and 'Last'.

116970 nolu istasyonun 1990 yılı ocak ayında kaydettiği en düşük, en yüksek ve ortalama sıcaklık değerlerini listelleyen sql cümlesini yazın.

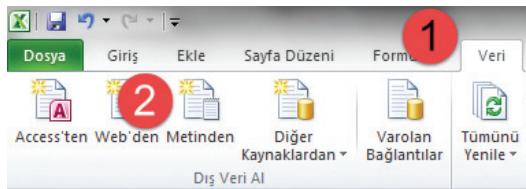


SIRA SİZDE

Verileri Microsoft Excel ya da başka bir yazılımda grafik olarak göstermek istersek verileri indirmemiz gereklidir. Eğer verilerimizi Microsoft Excel programında grafik olarak göstermek istersek Google sunucularında bulunan veriyi indirmek için "Download as CSV" düğmesini kullanmak en uygun çözüm olacaktır. Veriyi indirdikten sonra Microsoft Excel programını açıyoruz. Sırası ile "Veri" (1) menüsüne oradan da "Metinden" düğmesine (2) tıklıyoruz.

Resim 8.9

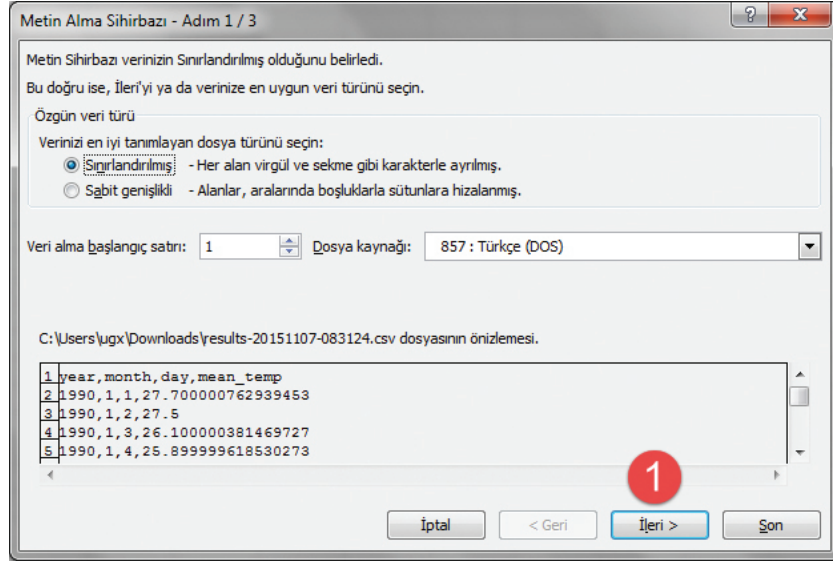
CSV Dosyasının  
Microsoft Excel  
programına  
Aktarılması.



Açılan ekranda hangi CSV dosyasını aktaracaksak o dosyayı seçiyoruz. Burada bizi üç adımlı bir sihirbaz karşılayacaktır, ilk adım Resim 8.10'da görünen seçili dosyanın ön izlemesidir. Bu ekranı geçmek için “İleri>” tuşuna basmanız gereklidir.

**Resim 8.10**

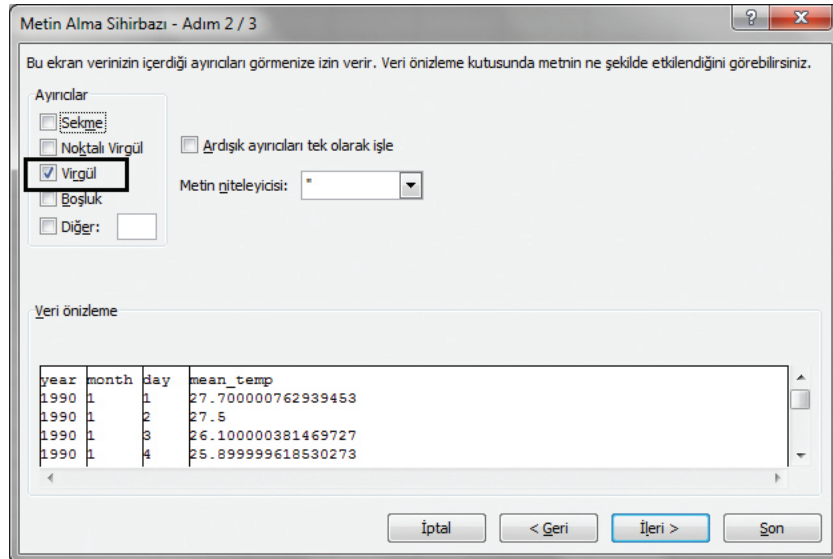
Metin Alma Sihirbazı  
Adım 1.



Aşağıda Resim 8.11'de görünen ikinci adımda virgül seçeneğini seçerek verilerin kolumnlara ayrılmasını sağlıyoruz. Sihirbazın bir sonraki aşamaya geçmesi için “İleri>” tuşuna basmak yeterlidir.

**Resim 8.11**

Metin Alma Sihirbazı  
Adım 2.



Adım 2'de “İleri>” tuşuna bastığımızda son adım olan üçüncü adıma geçeriz. Veri alma sihirbazının son adımı Resim 8.12'deki gibi görünmektedir.

Eğer işlemlerimizi doğru bir şekilde yapmışsak veriler sütunlar hâlinde gösterilecektir.



DİKKAT

Resim 8.12

Metin Alma Sihirbazı  
Adım 3.

Metin Alma Sihirbazı - Adım 3 / 3

Bu ekran her bir sütunu seçmenize ve Veri Biçimi'ni belirlemenize izin verir.

Sütun veri biçimi

☒ Genel  
☐ Metin  
☐ Tarih: GAY  
☐ Sütundan veri alma (atla)

'Genel', sayısal değerleri sayılara, tarih değerlerini tarihlere ve diğer tüm değerleri metne dönüştürür.

Gelişmiş...

Veri önizleme

year	month	day	mean_temp
1990	1	1	27.700000762939453
1990	1	2	27.5
1990	1	3	26.100000381469727
1990	1	4	25.899999618530273

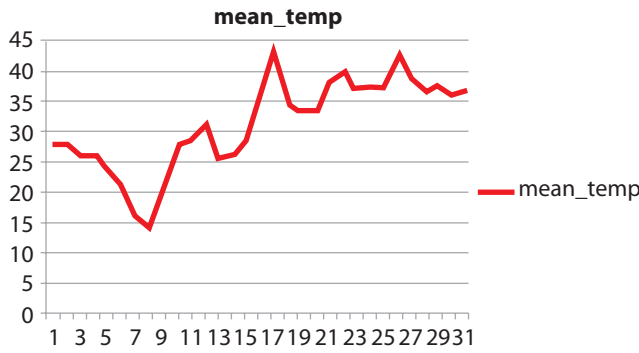
İptal < Geri İleri > Son

Son olarak “Son” tuşuna basarak verilerimizi Microsoft Excel programımıza yüklüyoruz. Artık tüm veriler elimizde olduğuna göre Microsoft Excel programını kullanarak istediğimiz grafiği çizdirebiliriz. Örneğin Resim 8.13’de 1990 yılının ocak ayının sıcaklık (Fahrenheit) grafiği görülmektedir.

**Fahrenheit** sıcaklık ölçü birimi Alman fizikçi Daniel Gabriel Fahrenheit tarafından 1724 yılında oluşturulan bir sıcaklık ölçüm birimidir. Amerika Birleşik Devletlerinde sıcaklıkların gösterimi için kullanılır.

Resim 8.13

1990 Yılı Ocak Ayı  
Ortalama Sıcaklık  
Grafiği.



2000 yılının şubat ayına ait gün/ ortalama sıcaklık grafiğini çiziniz.



SIRA SİZDE

3



## Özet



### *Bulut bilişimi kavramını tanımlamak.*

Bulut bilişim tek bir yerel sunucunun ya da kişisel bir cihazın kaynağına bağlı olarak yapılan hesaplama yerine birden fazla cihazın kaynaklarını kullanarak hesaplama yapma işi olarak tanımlanabilir. İnternet üzerinden sunulan bu ürünler hizmet olarak adlandırılır. Bulut bilişim erişim modellerine göre genel bulut, topluluk bulut, özel bulut ve karma bulut olmak üzere dörde ayrılır. Erişim modeli bulut yapısına kimlerin ve nereden ulaşacağını belirler. En genel kullanıcıya hitap eden genel bulut yapıları iken en özel kullanıcı grubuna hitap eden özel bulut yapılarıdır.



### *Bulut bilişim hizmet ve erişim modellerini açıklamak.*

Bulut bilişim kapsamında son kullanıcıya sunulan her ürün hizmet olarak adlandırılır. Sunulan hizmetler temel olarak üç ana başlık altında modellenir. Bu başlıklar hizmet olarak altyapı, hizmet olarak platform, hizmet olarak yazılım şeklinde sıralanabilir. Hizmet olarak altyapı modelinde son kullanıcıya ağ altyapısı, disk alanı gibi fiziksel ürünler sunulur. Hizmet olarak platform yapıları daha çok geliştiriciler için yazılım geliştirme ve geliştirilmiş olan yazılımın ölçeklendirilebilir şekilde çalıştırılması için ortam sağlar. Son olarak hizmet olarak yazılım modellerinde başka şirketler tarafından geliştirilmiş olan paket yazılımlar ve ya servisler son kullanıcıya dağıtılır. Her üç modelde de müşteriler, kullandığın kadar öde ya da ücretsiz kullanım modeli ile faturalandırılırlar.



### *NoSQL veritabanı yapılarını tanımlamak.*

Son yıllarda internet üzerinde paylaşılacak için kullanıcılar tarafından üretilen veri, her geçen gün ivmeLENEREK artmıştır. Büyüyen bu verinin analiz edilmesi, üzerinden sorgular yapılması ve verinin anlamlandırılması gibi problemlerin üstesinden gelmek için üreticiler mevcutta var olan ilişkisel veritabanı yerine 2000'li yılların başında ortaya çıkan NoSQL yapılarına yönelmişlerdir. Ancak bu kadar devasa bir veri üzerinde işlem yapmak için belirli bir çözüm yolu ve standart olmadığından şirketler belli başlı problemlerin çözümü için çeşitli NoSQL veritabanı yapıları ortaya koymuşlardır. Şirketlerin farklı alanlardaki çözümler için ürettikleri NoSQL yapılarını temelde çizge tabanlı, anahtar-değer tabanlı, sütun tabanlı ve belge tabanlı olarak dörde ayırabiliriz. Her bir grup NoSQL ürünü farklı alanlarda başarılı sonuçlar üretmektedirler.

## Kendimizi Sıyalım

1. Aşağıdaki özelliklerden hangisi Google BigQuery servisini oluşturan yapılardan biri **değildir**?
  - a. Görevler
  - b. Rest Api
  - c. Tablolar
  - d. Projeler
  - e. Veri Bloğu
2. Aşağıdaki durumlardan hangisi NoSQL kullanmak için en geçerli sebeptir?
  - a. Veri bütünlüğünün ve gizliliğinin çok kritik olduğu yapılarda
  - b. Hızlı sorgu cevabının önemsiz olduğu durumlarda
  - c. Büyük veri üzerinde sorgu yapılması gerektiği durumlarda
  - d. Veri homojen olarak dağıtıldığı durumlarda
  - e. Proje bütçesi çok büyük olduğunda
3. Twitter'dakine benzer büyüklükte ve birbirleri ile sıkı ilişkileri olan verileri tutmak için hangi tip veritabanı tercih edilmelidir?
  - a. Çizge tabanlı NoSQL veritabanı yapıları
  - b. Anahtar-Değer tabanlı NoSQL veritabanı yapıları
  - c. Sütun tabanlı NoSQL veritabanı yapıları
  - d. Belge Tabanlı NoSQL veritabanı yapıları
  - e. Klasik ilişkisel veritabanı
4. Aşağıdaki yapılardan hangisi Cassandra veritabanı veri modeli içerisinde **yer almaz**?
  - a. Sütun ailesi
  - b. Anahtar
  - c. Anahtar alanı
  - d. Sütun
  - e. İndex
5. Aşağıdakilerden hangisi NoSQL yapılarının özelliklerinden **biri değildir**?
  - a. Açık kaynak olması
  - b. Dağıtık yapılandırılabilir olması
  - c. Şema bağımsız bir yapıda olması
  - d. Tablolar arası ilişkiler belli alanlar üzerinden kurulması
  - e. Yatay olarak genişleyebilir olması
6. Aşağıdaki ürünlerden hangisi NoSQL veritabanlarına örnek **değildir**?
  - a. MySQL
  - b. MongoDB
  - c. CouchDB
  - d. Neo4J
  - e. Hadoop
7. Bulut bilişim ile ilgili aşağıdaki ifadelerden hangisi yanlıştır?
  - a. Esnek yapılıdır.
  - b. Kolay ölçeklendirilebilir.
  - c. Kullandığın kadar öde modeli ile faturalandırılır.
  - d. Değişik kullanıcı profilleri için farklı modeller önerebilir.
  - e. Yakın gelecekte kullanıcılar tarafından unutulacak bir teknolojidir.
8. Aşağıdakilerden hangisi bulut bilişim dağıtım modellerinden biri **değildir**?
  - a. Genel bulut
  - b. Hizmet olarak altyapı
  - c. Özel bulut
  - d. Karma bulut
  - e. Topluluk bulut
9. Amazon web servisleri bulut bilişimde hangi çeşit dağıtımdır?
  - a. Hizmet olarak yazılım
  - b. Özel bulut
  - c. Hizmet olarak altyapı
  - d. Genel bulut
  - e. Hizmet olarak platform
10. Bulut bilişimdeki “bulut” kelimesi hangi kavramı işaret etmektedir?
  - a. Kablosuz bağlantı
  - b. Sonsuz kaynak
  - c. Sabit disk
  - d. İnternet
  - e. Sonsuz hesap yeteneği

## Kendimizi Sınavalım Yanıt Anahtarı

1. b Yanıtınız yanlış ise “Google BigQuery servisi nasıl kullanılır” konusunu yeniden gözden geçiriniz.
2. c Yanıtınız yanlış ise “NoSQL Yapıları” konusunu yeniden gözden geçiriniz.
3. a Yanıtınız yanlış ise “Çizge Tabanlı NoSQL yapıları” konusunu yeniden gözden geçiriniz.
4. e Yanıtınız yanlış ise “Sütun Tabanlı NoSQL yapıları” konusunu yeniden gözden geçiriniz.
5. d Yanıtınız yanlış ise “NoSQL Yapıları” konusunu yeniden gözden geçiriniz.
6. a Yanıtınız yanlış ise “NoSQL Yapıları” konusunu yeniden gözden geçiriniz.
7. e Yanıtınız yanlış ise “Bulut Bilişim” konusunu yeniden gözden geçiriniz.
8. b Yanıtınız yanlış ise “Bulut Bilişim Dağıtım Modelleri” konusunu yeniden gözden geçiriniz.
9. c Yanıtınız yanlış ise, “Bulut Bilişim Hizmet Modelleri” konusunu yeniden gözden geçiriniz.
10. d Yanıtınız yanlış ise “Bulut Bilişim ” konusunu yeniden gözden geçiriniz.

## Sıra Sizde Yanıt Anahtarı

### Sıra Sizde 1

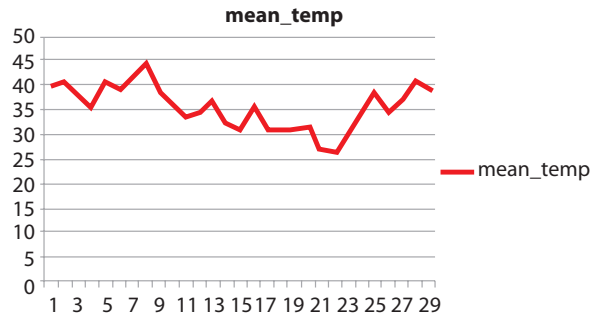
```
{
  OgrenciAd: 'Ahmet',
  OgrenciSoyad: 'Kocaman',
  OgrenciSinif: '1.Sınıf',
  Yorum: [
    {
      Kaydeden: 'Uğur GÜREL',
      Yorumİcerik: 'Çok başarılı bir öğrenci',
      Tarih: new Date(2015,11,2,2,35),
    }
  ]
},
{
  OgrenciAd: 'Ayşe',
  OgrenciSoyad: 'Güzelyurt',
  OgrenciSinif: '1.Sınıf',
  Yorum: [
    {
      Kaydeden: 'Uğur GÜREL',
      Yorumİcerik: 'Müzik konusunda çok yetenekli',
      Tarih: new Date(2015,10,22,10,22),
    }
  ]
}
```

### Sıra Sizde 2

```
SELECT
  AVG(mean_temp) as Ortalama,
  MIN(mean_temp) as Minimum,
  MAX(mean_temp) as Maximum
FROM
  [publicdata:samples.gsod]
WHERE
  station_number=116790
  AND year = 1990
  AND month = 1
```

### Sıra Sizde 3

Oluşması gereken grafik aşağıda gösterilmiştir.



## Yararlanılan ve Başvurulabilecek Kaynaklar

- Fowler A., (2015). **NoSQL For Dummies**, Wiley and Sons.  
<http://dist.neo4j.org/basic-neo4j-code-examples-2008-05-08.pdf> (Son erişim 06.11.2015)  
<https://www.linkedin.com/pulse/choosing-right-NoSQL-database-type-app-development-snehesh-mitra> (Son erişim 06.11.2015)  
<https://cloud.google.com/bigquery/> (Son erişim 06.11.2015)  
Özsu, M. T., ve Valduriez, P. (2011). **Principles of Distributed Database Systems**. Springer Science & Business Media.  
Sato, K. (2012). **An Inside Look at Google BigQuery, White Paper**. Google Inc.