VERİ TABANI VE YÖNETİM SİSTEMLERİ

Dr. Öğretim Üyesi

Alper Talha KARADENİZ

GENEL DERS İÇERİĞİ

1. NoSQL Nedir?
2. ACID-BASE
3. CAP Teoremi
4. NoSQL Kullanım Alanları
5. RDBMS

NoSQL Nedir?

NoSQL (Not Only SQL), ilişkisel olmayan veri tabanı yönetim sistemlerini tanımlar.
Geleneksel SQL tabanlı ilişkisel veritabanlarının aksine, NoSQL sistemleri esnek veri modelleri, yüksek ölçeklenebilirlik, yüksek performans ve dağıtık mimari özellikleri ile dikkat çeker.

Verileri tablo-satır-sütun formatında değil; genellikle anahtar-değer, belge (JSON, BSON), graf ya da sütun temelli biçimlerde saklar.

NoSQL’e Neden İhtiyaç Duyuldu?

NoSQL veritabanları, geleneksel ilişkisel veritabanlarının belirli durumlarda yetersiz kalması ve yeni nesil uygulama ihtiyaçlarının artmasıyla ortaya çıkmıştır.
İşte bazı nedenler:

1. Büyük Veri (Big Data) İhtiyacı

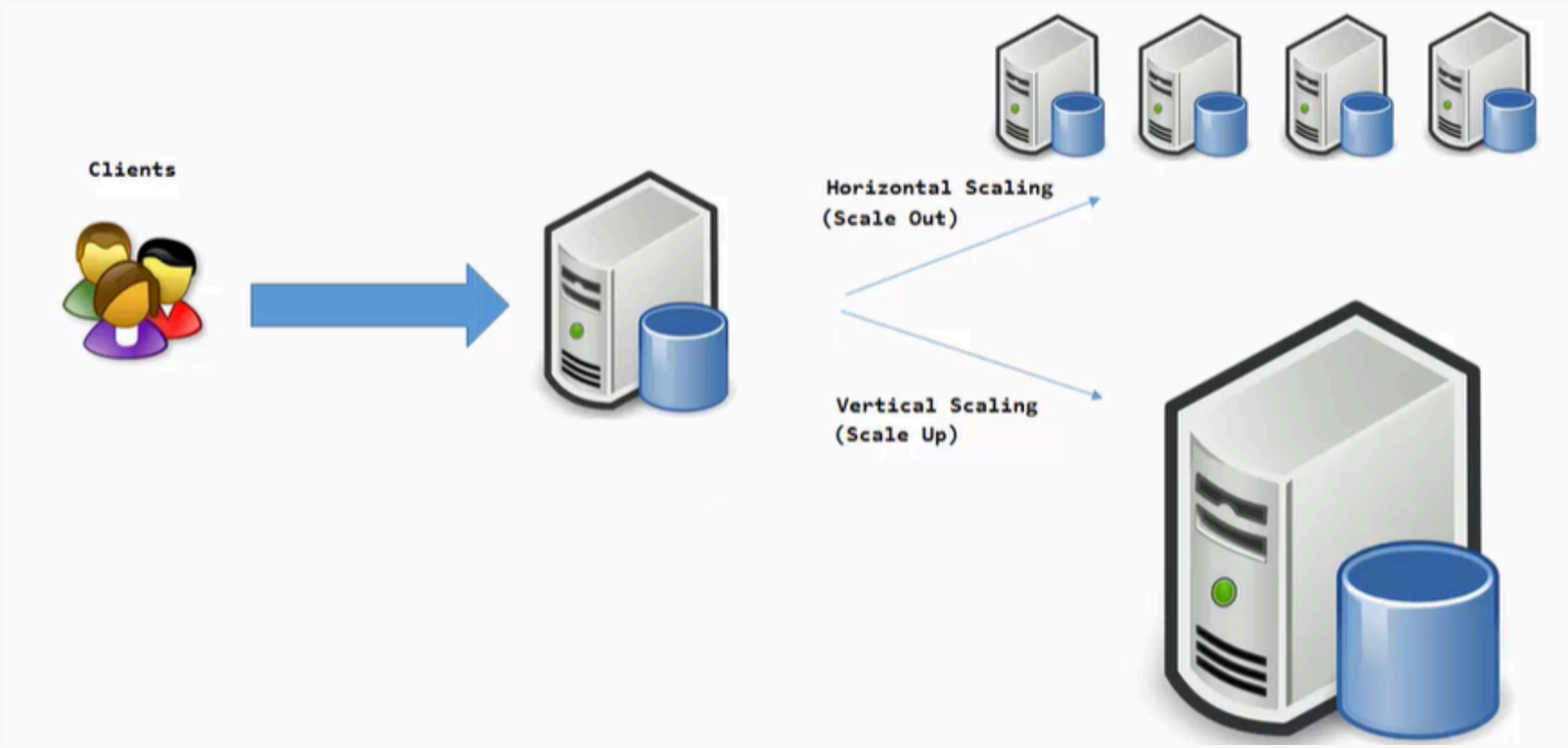
İnternetin ve dijitalleşmenin yaygınlaşmasıyla veri hacmi katlanarak arttı. RDBMS sistemleri bu kadar büyük ve hızlı veri akışını verimli şekilde işleyemez hâle geldi. NoSQL sistemler, yatay ölçeklenebilirlik sayesinde büyük veriyi parçalayarak yönetebildi.

NoSQL’e Neden İhtiyaç Duyuldu?

2. Ölçeklenebilirlik Sorunları

RDBMS’ler dikey ölçeklenir (daha güçlü sunucu gerekir), bu da maliyetlidir.

NoSQL ise yatay ölçeklenir, yani sisteme yeni sunucu eklemek yeterlidir. Daha pratik ve uygun fiyatlı olması nedeniyle tercih edilir.



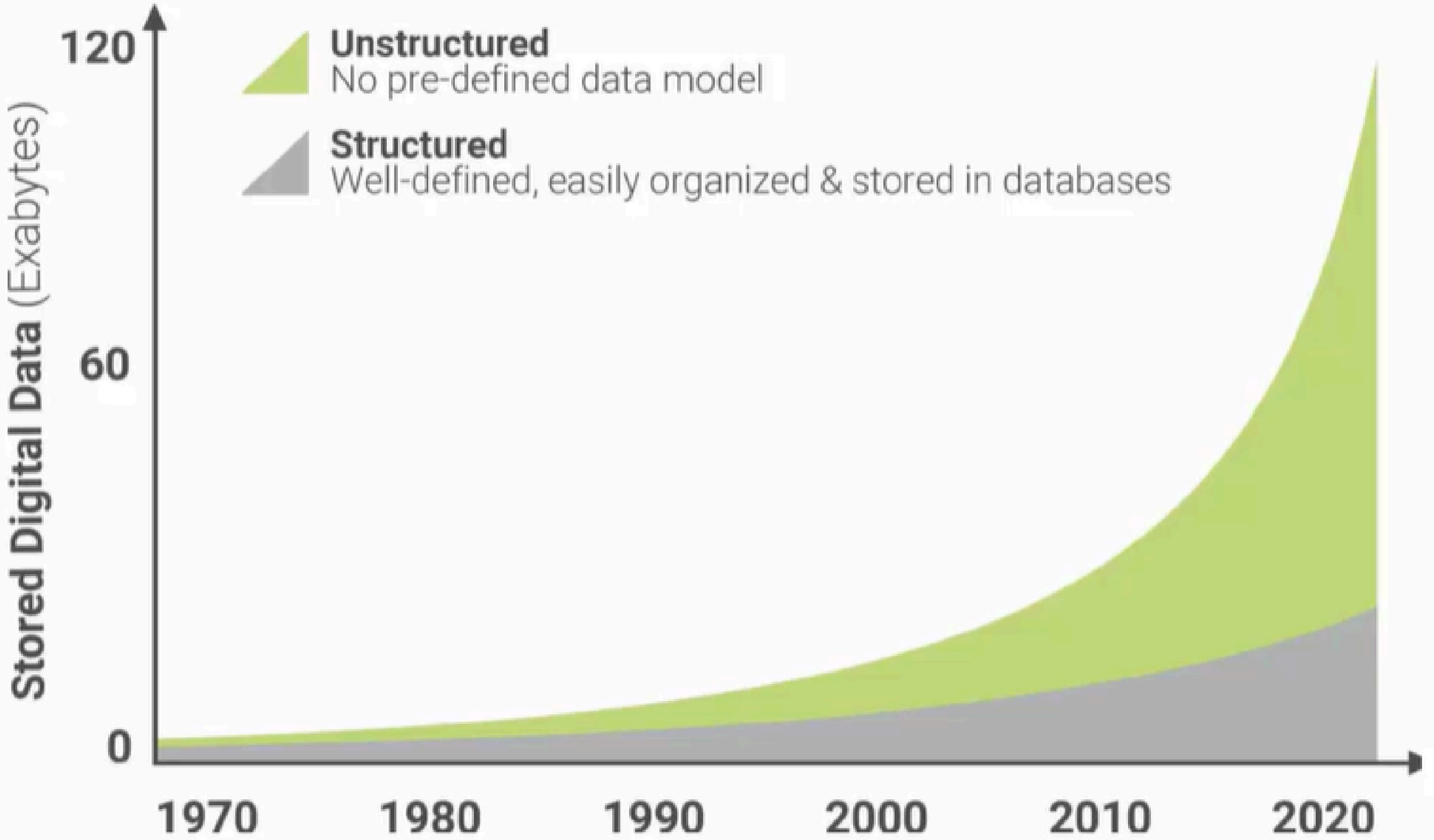
NoSQL’e Neden İhtiyaç Duyuldu?

3. Veri Çeşitliliği

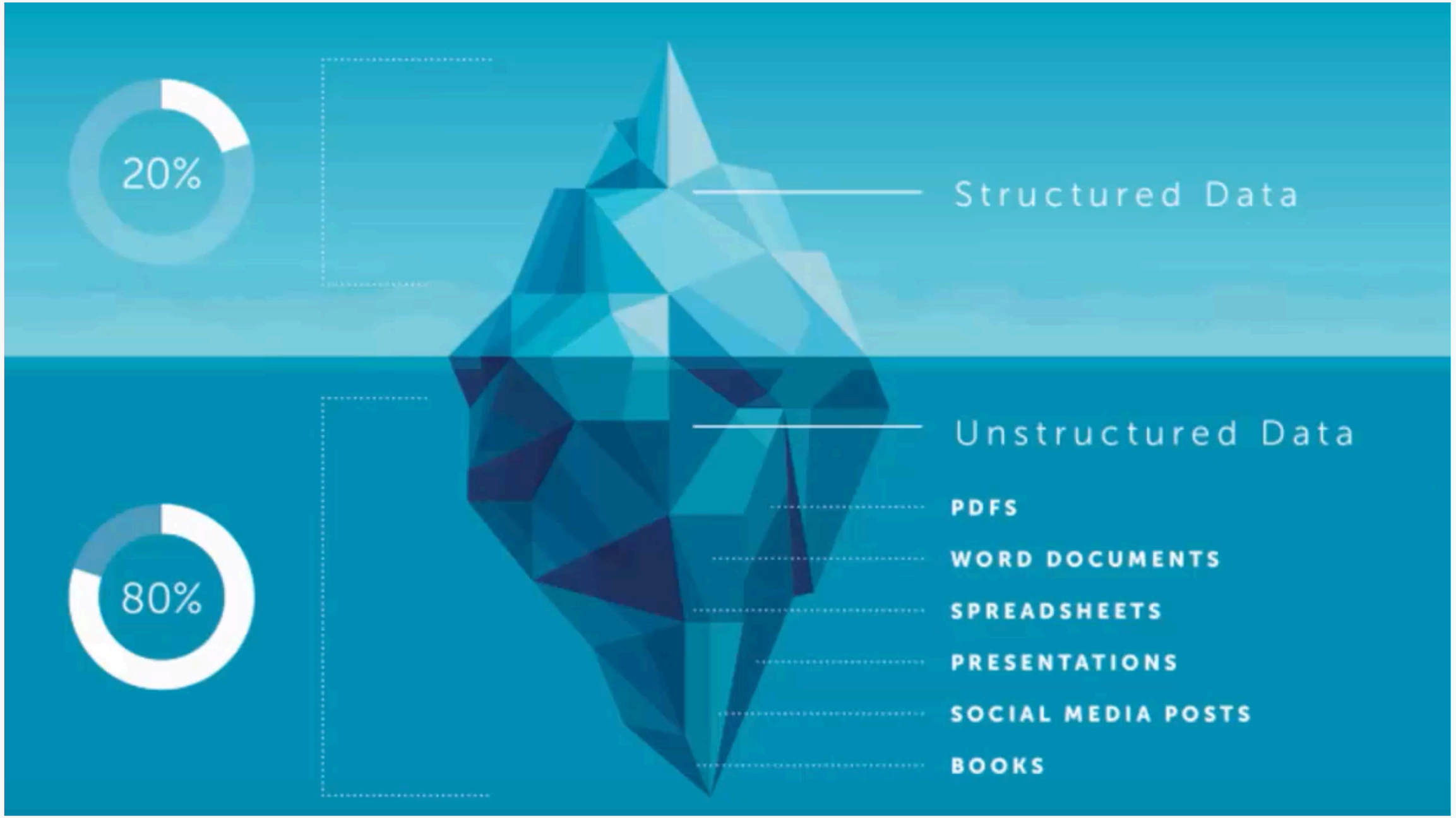
Günümüzde veriler sadece tablo biçiminde değil; JSON, XML, metin, video, log dosyası, sensör verisi gibi çok farklı formatlarda gelmektedir. Bu çeşitlilik, geleneksel ilişkisel veritabanlarında yönetilmesi zor hale gelen karmaşık veri türlerini ortaya çıkarmıştır.

Veri türleri genel olarak üçe ayrılır:

- ***Yapılandırılmış (Structured) Veri:*** Satır-sütun düzenine sahip, belirli bir şemaya göre organize edilen verilerdir. Örneğin; müşteri tablosu (ID, Ad, Soyad).
- ***Yarı Yapılandırılmış (Semi-Structured) Veri:*** Katı bir tablo düzeni yoktur ancak veri içinde etiketler ya da anahtar-değer çiftleri vardır. Örneğin; JSON, XML belgeleri.
- Y***apılandırılmamış (Unstructured) Veri:*** Herhangi bir belirli düzeni olmayan, analiz için özel işlem gerektiren verilerdir. Örneğin; videolar, ses dosyaları, e-postalar, sosyal medya içerikleri.



Data Variety – Veri Çeşitliliği

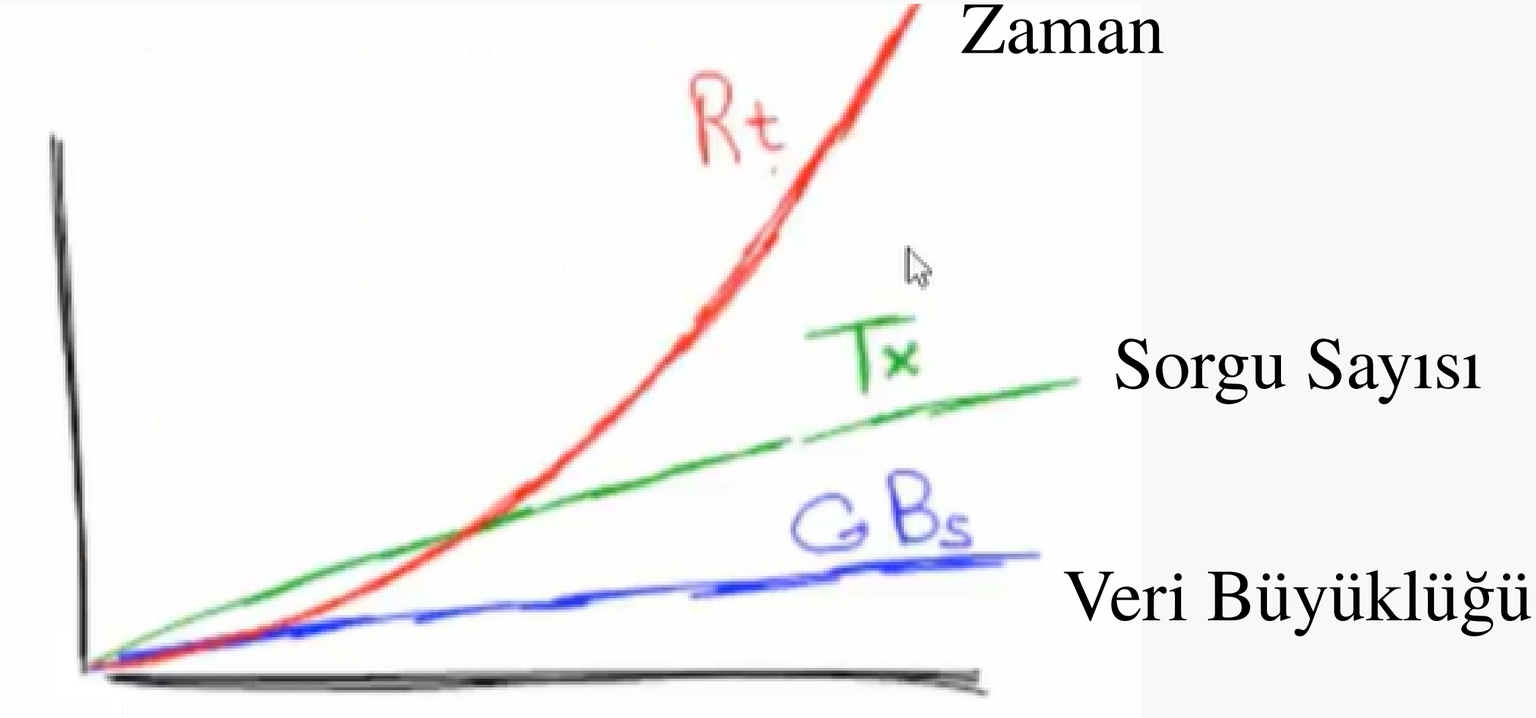

20%
Structured Data
80%
Unstructured Data
PDFS
WORD DOCUMENTS
SPREADSHEETS
PRESENTATIONS
SOCIAL MEDIA POSTS
BOOKS

NoSQL’e Neden İhtiyaç Duyuldu?

4. Yüksek Performans ve Hız İhtiyacı

Gerçek zamanlı uygulamalar (örneğin: sosyal medya, oyunlar, canlı analiz sistemleri) milisaniyelik yanıt süreleri gerektirir.

NoSQL sistemler, önceden tanımlı yapılar ve bellek tabanlı işlemler sayesinde bu hızı sağlar.



ACID & BASE

ACID, veritabanı işlemlerinin güvenilirliğini sağlamak için kullanılan dört temel özelliği ifade eder. Dikey ölçeklendirme kullanılır.
Bankacılık gibi yüksek doğruluk ve güvenlik gerektiren uygulamalarda tercih edilir.

Atomicity (Atomiklik): Bütünlük. Bir işlem ya tamamen yapılır ya da hiç yapılmaz. Parçalı başarıya izin verilmez.
Consistency (Tutarlılık): Tutarlılık. İşlem öncesi ve sonrası veritabanı geçerli bir durumda olmalıdır. Yani mevcut veri yeni değeri alabilmeli.
Isolation (Yalıtım): Tek işlem. Aynı anda çalışan işlemler birbirini etkilemeden yürütülmelidir. İlgili veri kilitlenir ve işlemler seri halde yapılır.
Durability (Kalıcılık): Dayanıklılık. İşlem başarıyla tamamlandıysa, sonucu kalıcıdır, sistem çökse bile kaybolmaz.

ACID & BASE

BASE, ACID'in tam tersi değildir, ancak daha esnek bir yaklaşım sunar. Özellikle dağıtık sistemlerde hız ve ölçeklenebilirlik ön plandadır. Yatay ölçeklendirme kullanılır.

Sosyal medya, gerçek zamanlı analiz, büyük ölçekli sistemler gibi yüksek hız ve esneklik isteyen durumlarda kullanılır.

Basically Available (Temel Erişilebilirlik): Sistem çoğunlukla erişilebilir olur, her zaman tam tutarlılık garantilenmez.

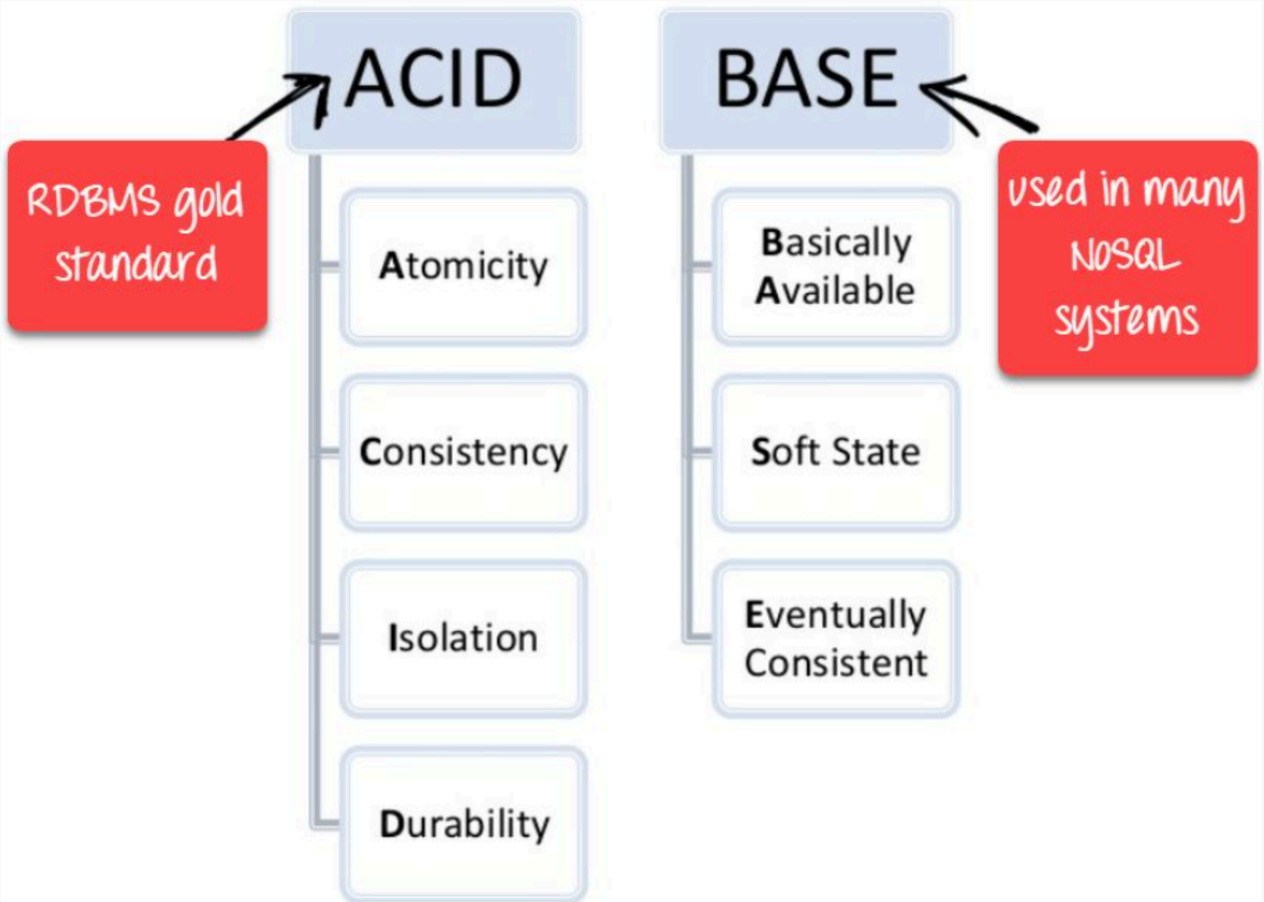
Soft state (Geçici Durum): Sistem durumu zamanla değişebilir, işlemler anında kesin sonuç vermez.

Eventually Consistent (Sonunda Tutarlı): Tüm veriler zaman içinde tutarlı hale gelir, ancak anlık tutarlılık şart değildir.

ACID & BASE

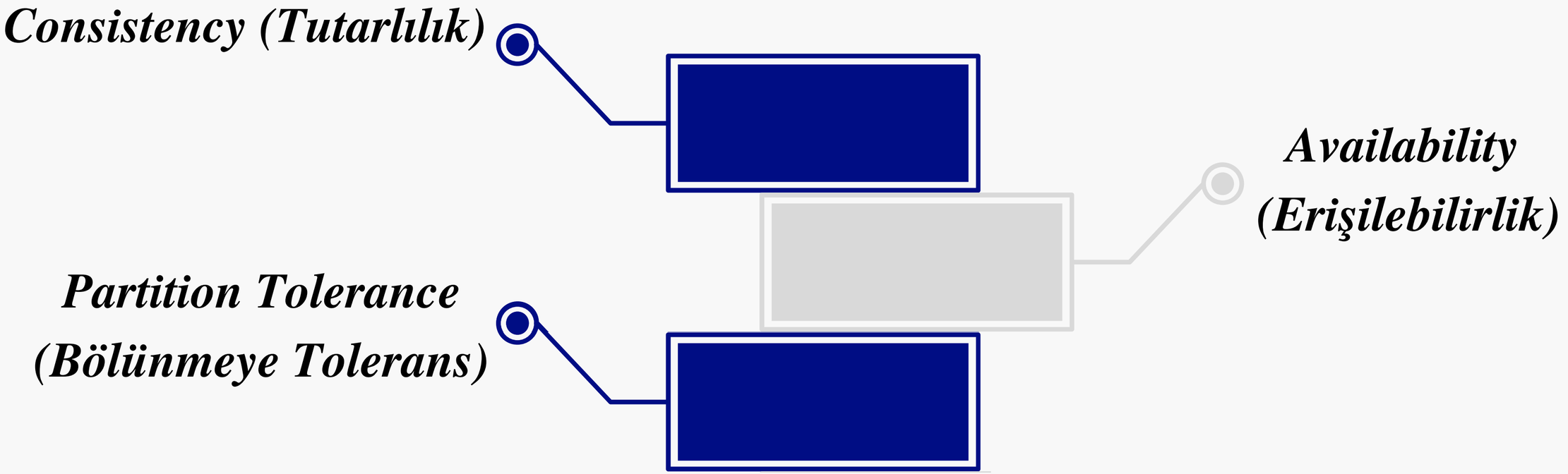
ACID : Güvenlik, doğruluk ve tutarlılık → Genelde SQL veritabanları için.

BASE : Esneklik, performans ve ölçeklenebilirlik → Genelde NoSQL veritabanları için.



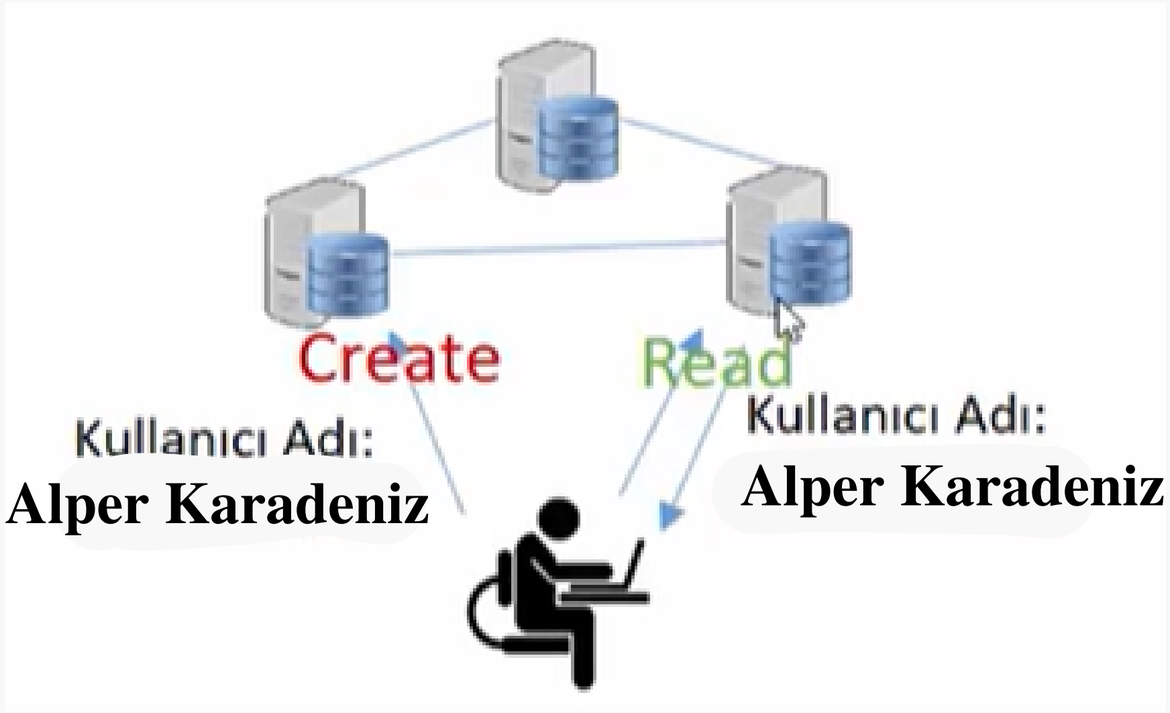
CAP Teoremi

Bir veritabanı sisteminin aynı anda üç temel özelliği birden tam olarak sağlayamayacağını ifade eder.



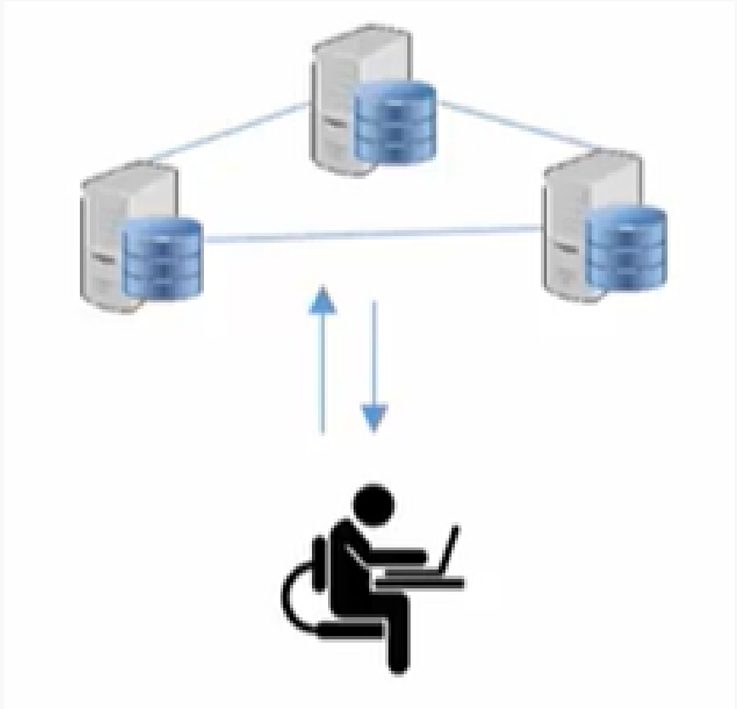
Consistency (Tutarlılık)

Tüm istemciler her zaman aynı veriyi görür. Dağıtık sistemde bir veri güncellendiyse, tüm düğümler hemen bu değişikliği yansıtmalıdır.
Örneğin DS'de bir sunucuya Kullanıcı Adı: Alper Karadeniz yazdık. Başka bir sunucuya Kullanıcı Adı: ? sorgusu attığımızda bize Alper Karadeniz cevabını veriyorsa buna tutarlılık denir.



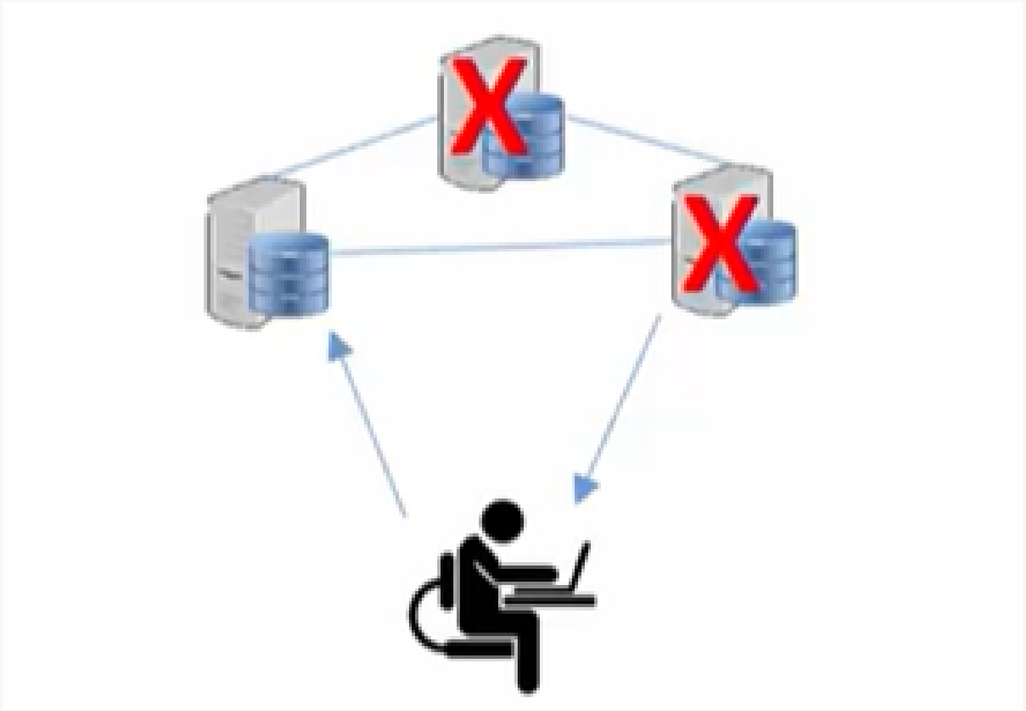
Availability (Erişilebilirlik)

Her istek, bir yanıt almalıdır (başarılı ya da başarısız, ama zamanında bir yanıt döner).



Partition Tolerance (Bölünmeye Tolerans)

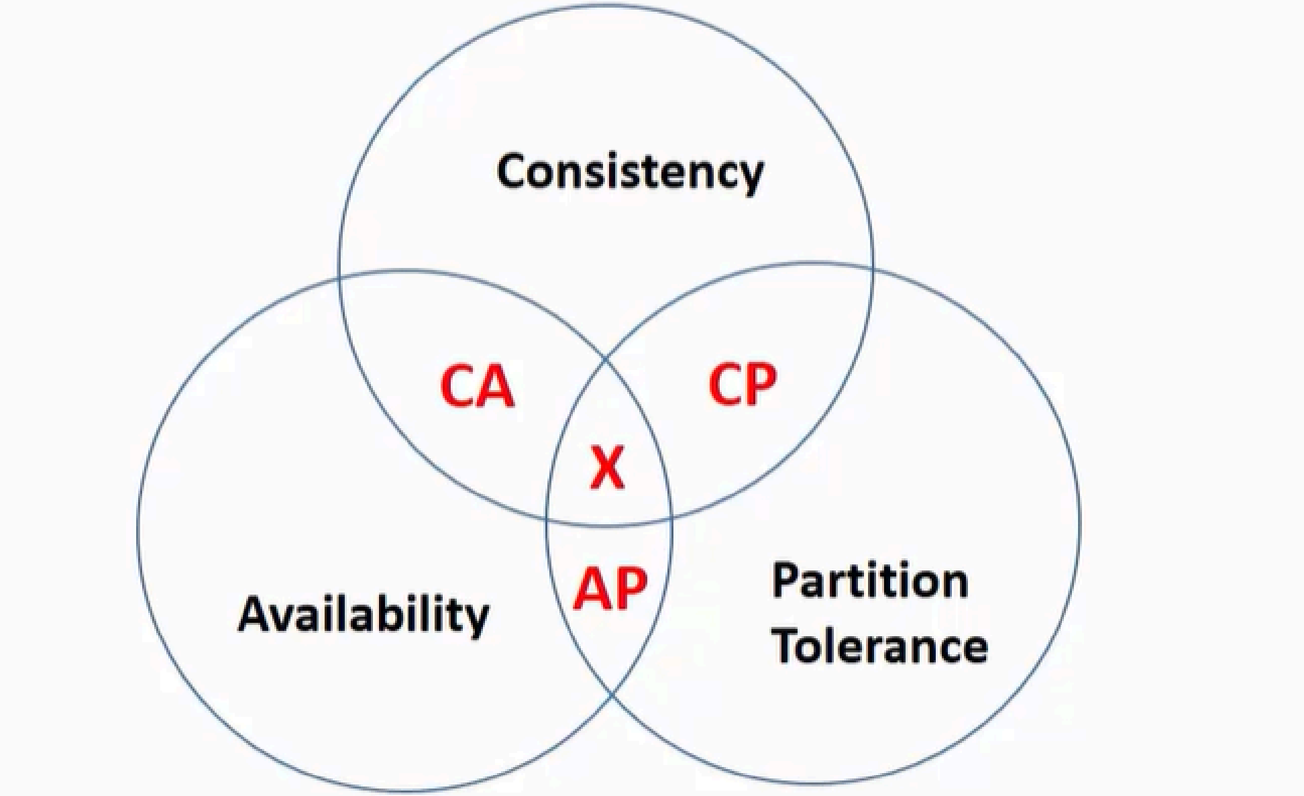
Sistem, ağ bağlantısı bozulsa bile çalışmaya devam etmelidir. Yani bazı düğümler birbiriyle iletişim kuramazsa bile sistem çökmez.



CAP Teoremi

Bir dağıtık sistem, aynı anda bu üç özelliği birden tam olarak sağlayamaz.

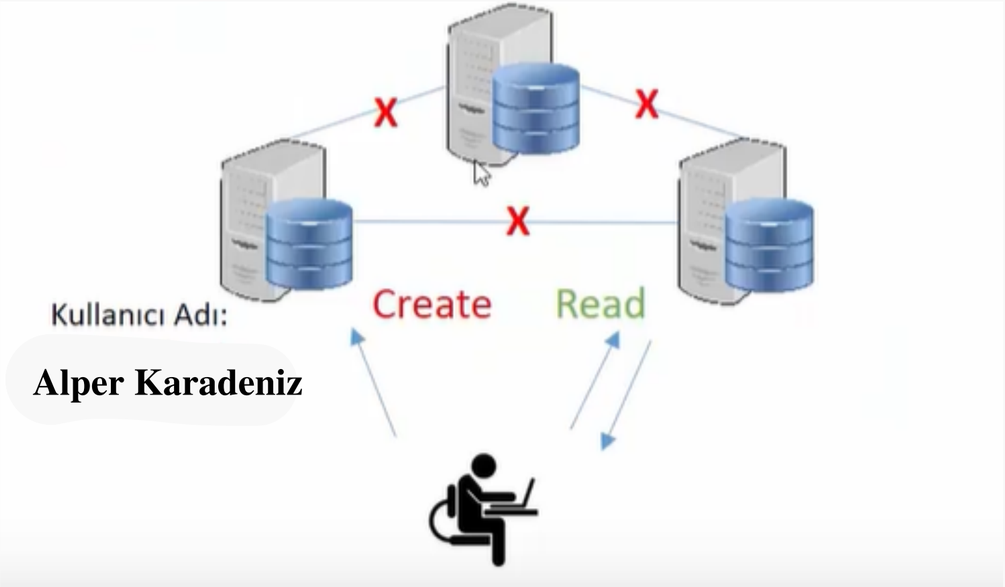
Yani bir dağıtık veritabanı yalnızca ikisini tam sağlayabilir, üçüncüsünden ödün vermek zorundadır.



CA’da (Consistency + Availability) Neden P Yok?

Clusterdaki sunucularımızda ağ bağlantı sorunu yaşanmış. Yani sunucularımız arasındaki bağ çözülmüş. Bir veri geldi ve bir numaralı sunucuya yazıldı. Diğer sunuculara da veriyi yazması gerekiyor çünkü Consistency istiyoruz. Fakat yazamıyor, bağlantı yok. Diğer sunuculara sorgu gönderdiğimizde veriyi bulamayacak.

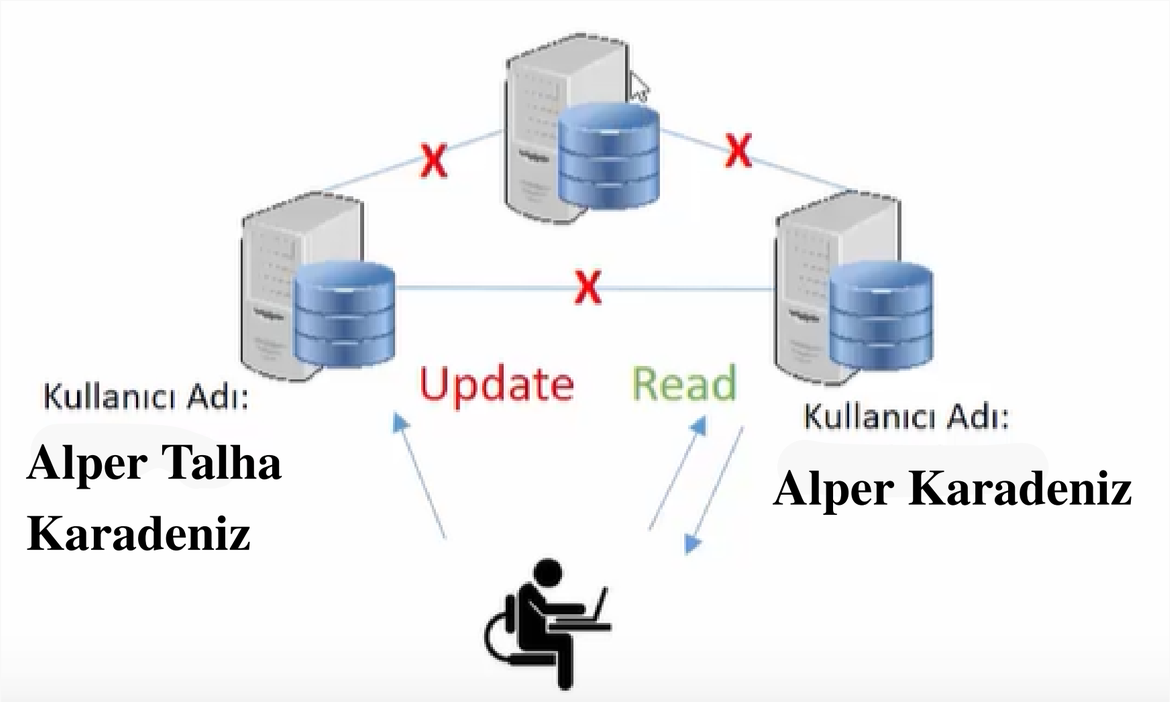
Hem verinin tutarlı olmasını hem de her zaman erişilebilir olmasını istediğimizde Bölme Toleransını sağlayamıyoruz.



AP’de (Availability + Partition Tolerance) Neden C Yok?

3 sunuculu bir cluster’da tüm sunucularda Kullanıcı Adı: Alper Karadeniz verisi bulunuyor. Yine sunucular arasında ağ bağlantı sorunu yaşanıyor. 1. sunucuda Kullanıcı Adı: Alper Talha Karadeniz olarak güncellendi. 2. sunucu üzerinde sorgu attığımızda bağlantı kaybolduğu için orada eski veri var. Bize sonuç olarak Kullanıcı Adı: Alper Karadeniz dönecektir.

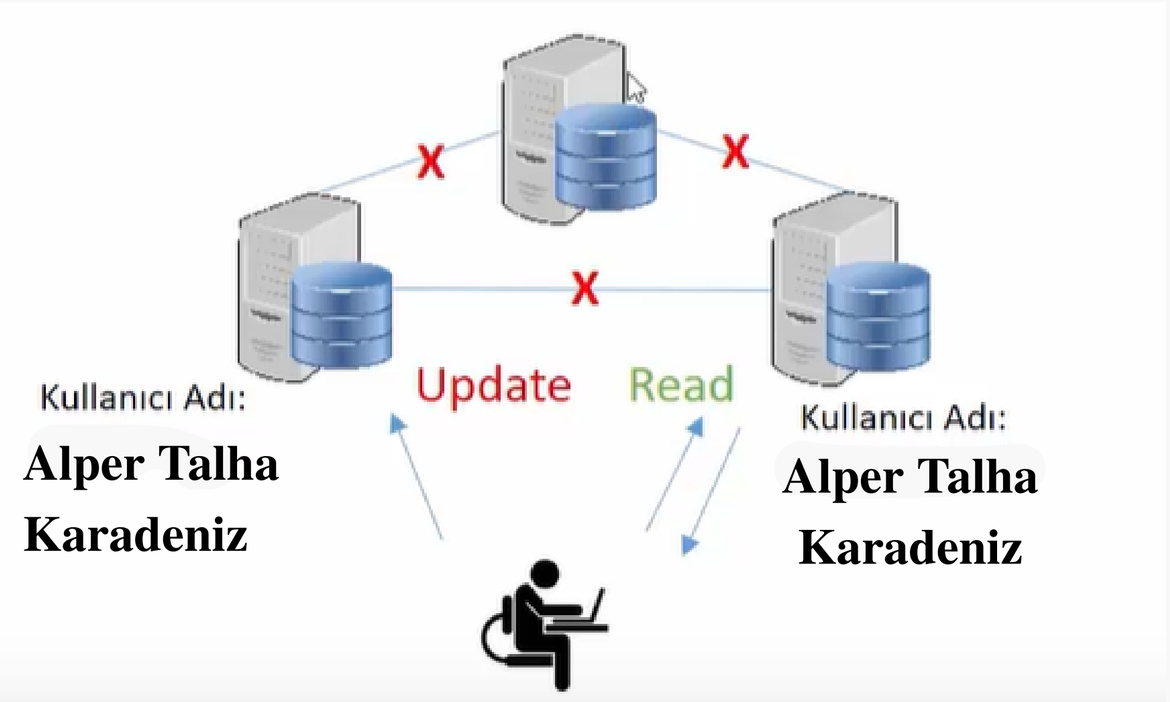
Bu durumda Consistency olmaz.



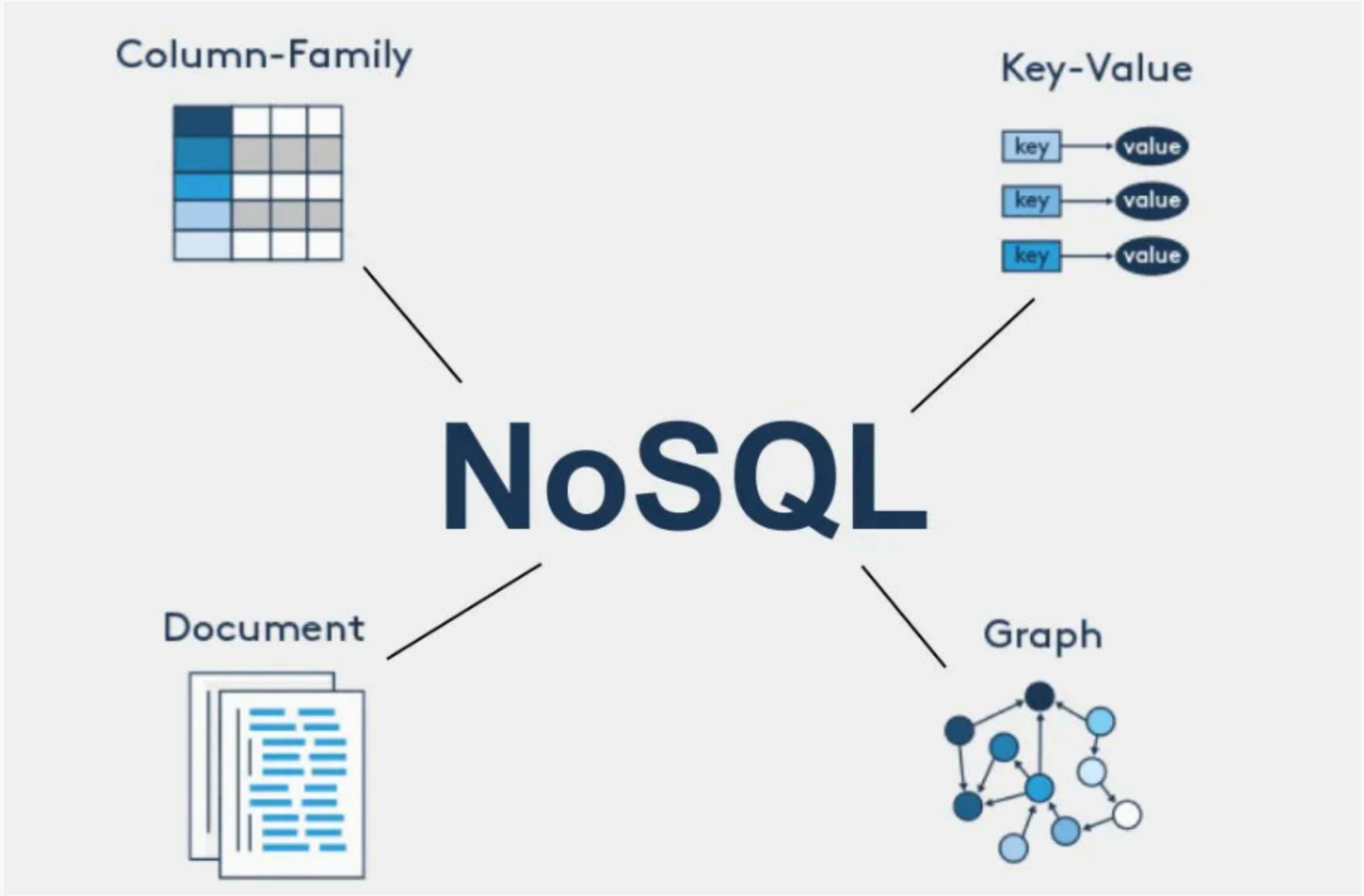
CP’de (Consistency + Partition Tolerance) Neden A Yok?

Sistemin hem tutarlı hem de verinin network üzerinde dağıtılmasını istiyoruz. Fakat cluster’da bir sunucuda bağlantı koptuğu an erişilebilirlik ortadan kalkacak. Kopan sunucuyu kapatmalısınız yoksa tutarlılık bozulacaktır.

MondoDB veritabanı gibi sistemler CP uygundur.
Default’ta strongly consistent’dır.



NoSQL Veri Türleri



Column-Family (Sütun Tabanlı)

- Veriler, satır yerine sütun grupları (column families) hâlinde saklanır.
- Aynı sütuna ait veriler fiziksel olarak birlikte saklanır → veri erişimi optimize edilir.
- Büyük verilerde hızlı okuma/yazma işlemleri için tasarlanmıştır.

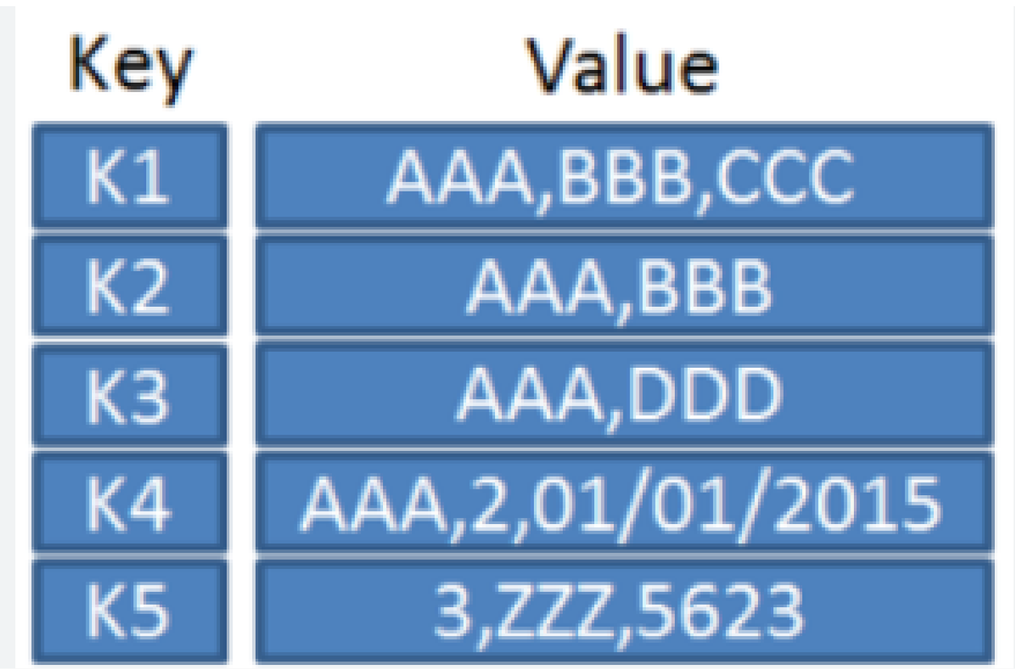
ID	Name	Grade	GPA
001	John	Senior	4.00
002	Karen	Freshman	3.67
003	Bill	Junior	3.33

Kullanım Alanları:

- Veri ambarları
- Log sistemleri
- Zaman serileri (time-series)
- Analitik uygulamalar

Key-Value (Anahtar-Değer)

- Veriler anahtar (key) ve değer (value) çiftleri şeklinde saklanır.
- Tıpkı bir sözlük (dictionary) ya da hash tablosu gibi çalışır.
- En basit ve en hızlı NoSQL veri modelidir.

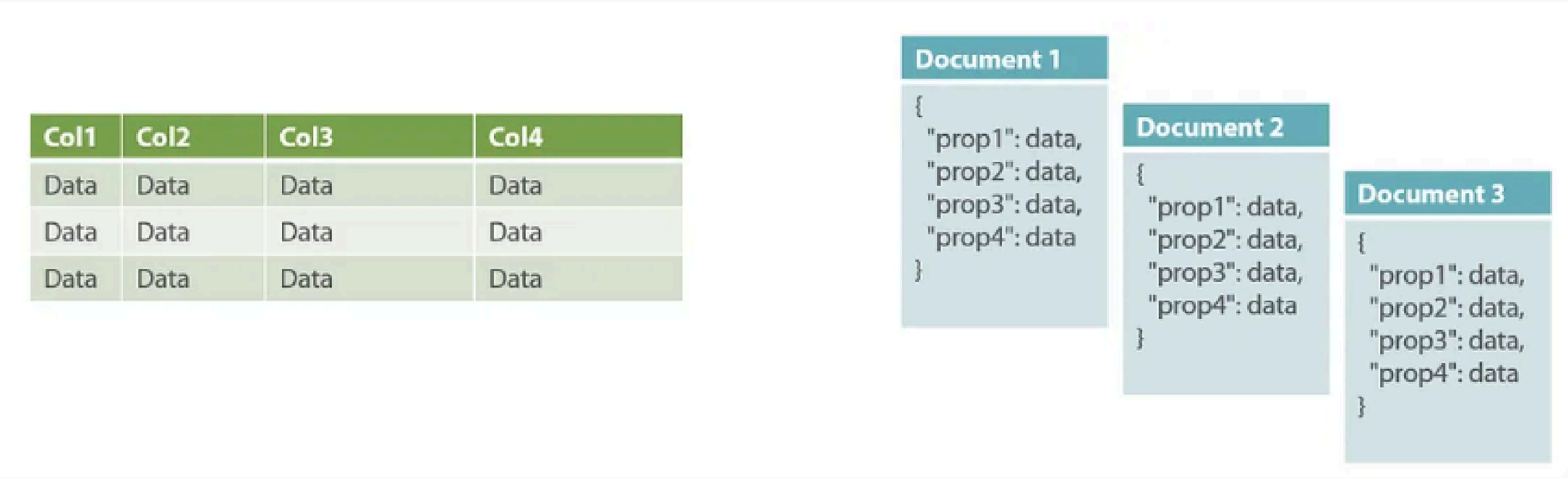


Kullanım Alanları:

- Önbellekleme (caching)
- Oturum yönetimi (session storage)
- Anlık veri erişimi gereken sistemler

Document (Belge Tabanlı)

- Veriler, genellikle JSON, BSON veya XML formatında saklanır.
- Her bir kayıt (document) anahtar-değer çiftlerinden oluşan bir belgedir.
- Her belge kendi şemasına sahip olabilir → esnek yapı sağlar.



Document (Belge Tabanlı)

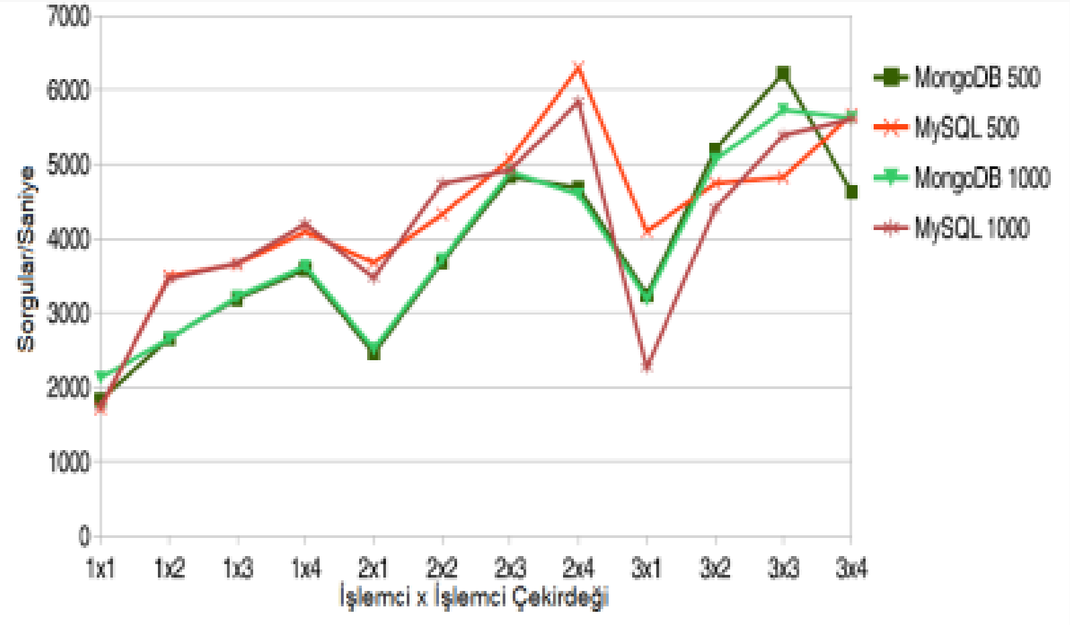
Kullanım Alanları:

- İçerik yönetim sistemleri (CMS)
- Blog, haber siteleri
- E-ticaret ürün katalogları
- Mobil uygulama verileri

Rank			DBMS	Database Model	Score		
Mar 2024	Feb 2024	Mar 2023			Mar 2024	Feb 2024	Mar 2023
1.	1.	1.	MongoDB	Document, Multi-model	424.53	+4.18	-34.25
2.	2.	2.	Amazon DynamoDB	Multi-model	77.72	-5.18	-3.05
3.	3.	3.	Databricks	Multi-model	74.34	-2.57	+13.48
4.	4.	4.	Microsoft Azure Cosmos DB	Multi-model	30.39	-1.60	-5.71
5.	5.	5.	Couchbase	Document, Multi-model	19.15	-1.33	-4.21
6.	6.	6.	Firebase Realtime Database	Document	15.07	-1.27	-3.71
7.	7.	7.	CouchDB	Document, Multi-model	11.73	-0.95	-2.73
8.	8.	8.	Google Cloud Firestore	Document	9.97	-0.90	-1.39
9.	9.	↑ 10.	Realm	Document	7.70	-0.26	-0.82
10.	10.	↓ 9.	MarkLogic	Multi-model	7.08	-0.42	-1.79

Graph (Graf Tabanlı)

- Veriler, düğümler (nodes) ve ilişkiler (edges) şeklinde modellenir.
- İnsan beyni gibi ilişkisel düşünür → veri noktaları arasındaki bağlantılar öne çıkar.
- Çok ilişkili verilerde (relationship-heavy data) en ideal sistemdir.



Kullanım Alanları:

- Sosyal ağlar
- Öneri sistemleri (recommendation engines)
- Dolandırıcılık tespiti (fraud detection)
- Organizasyon şemaları, yol haritaları

NoSQL Kullanım Alanları

1. Gerçek Zamanlı Web ve Mobil Uygulamalar: Düşük gecikme süresi, esnek veri yapıları, hızlı okuma/yazma özellikleri ile mesajlaşma uygulamaları, sosyal medya platformları, canlı skor uygulamalarında sıklıkla tercih edilir.

2. Büyük Veri (Big Data) İşleme: Yatay ölçeklenebilirlik, Hadoop gibi büyük veri platformlarıyla entegrasyon sayesinde milyonlarca satır veriyle çalışan analiz sistemlerinde (örneğin loglar, sensör verileri) tercih edilir.

3. E-Ticaret Sistemleri: Şemasız yapı ile farklı ürün türlerini esnek biçimde saklanabildiği için ürün katalogları, stok bilgileri, müşteri sepetleri, yorumlar gibi birçok alanda kullanılır.

NoSQL Kullanım Alanları

4. Kişiselleştirme ve Öneri Sistemleri: İlişki ağı (graf veri yapısı) ve hızlı veri erişimi ile film/dizi, müzik, ürün önerileri gibi alanlarda karşımıza çıkar.

5. Gerçek Zamanlı Analitik Sistemler: Hızlı veri girişi ve zaman serisi veri desteği sayesinde anlık kullanıcı davranış analizi, tıklama takibi, trafik analizi yapılabilir.

6. İçerik Yönetim Sistemleri (CMS): Esnek belge yapısı sayesinde farklı içerik türlerini kolayca tutabildiği için bloglar, haber siteleri, doküman yönetiminde tercih edilir.

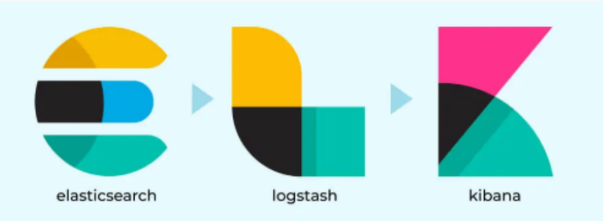
7. Konum ve Harita Uygulamaları: Yüksek hacmi sayesinde kullanıcı konum verileri, harita verisi, rota önerilerini rahatlıkla yapabilir.

NoSQL Kullanım Alanları

8. IoT (Nesnelerin İnterneti) Uygulamaları: Zaman serisi verileri depolama ve işlemeyi kolaylaştırır. Sensör verileri, cihaz izleme sistemlerinde kullanılır.

9. Oturum Yönetimi ve Kimlik Doğrulama: Anahtar-değer yapısı hızlı erişim sağlar. Kullanıcı giriş/çıkış oturumları, JWT token saklama

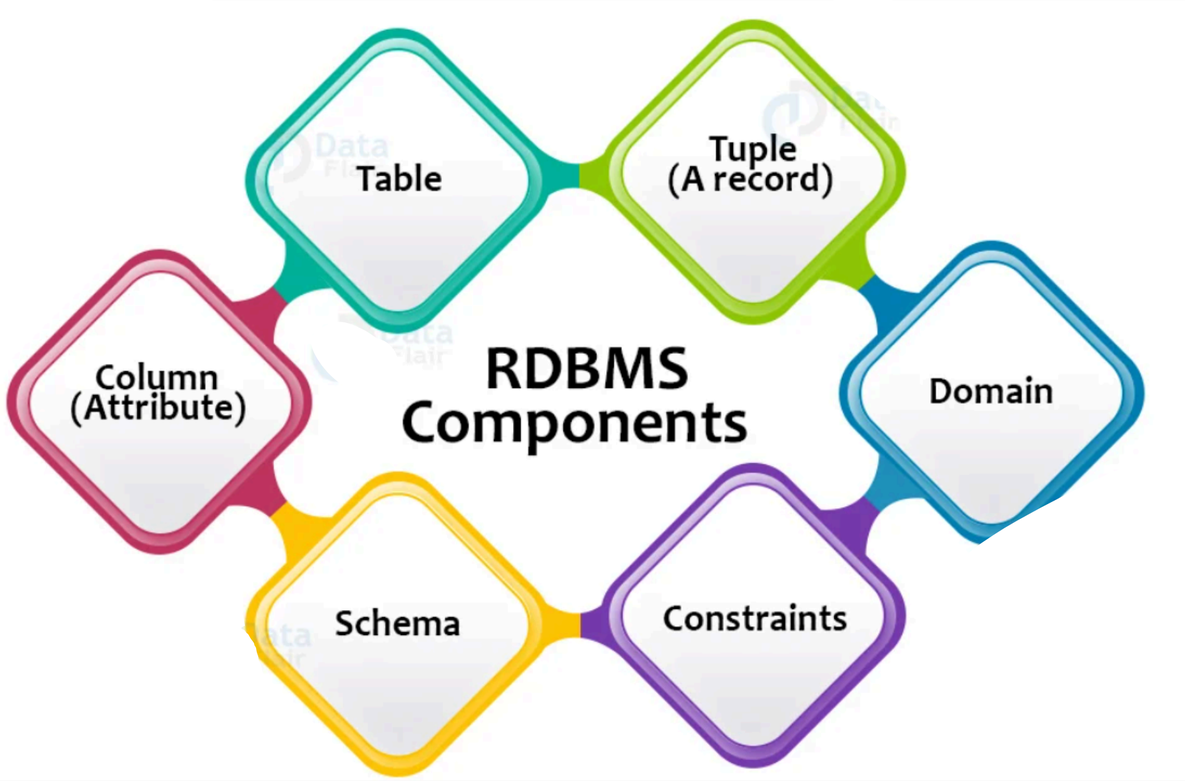
10. Log ve Olay Takip Sistemleri: Saniyede binlerce log verisini işlemek mümkün olduğu için uygulama hataları, kullanıcı eylemleri, güvenlik olaylarında sıklıkla tercih edilir.



RDBMS (Relational Database Management System)

RDBMS, verileri tablolar (table) şeklinde saklayan ve bu tablolar arasındaki ilişkileri yöneten veritabanı sistemidir.

Veriler satır (row) ve sütun (column) biçiminde organize edilir. Her tabloya ait bir şema (schema) tanımı vardır ve bu yapı önceden belirlenir.



RDBMS (Relational Database Management System)

Avantajları:

- Verilerin standart ve düzenli saklanması
- Veri tutarlılığı ve bütünlüğü
- SQL ile güçlü sorgulama yeteneği
- Karmaşık ilişkisel verilerin yönetimi
- Transaction (işlem) desteği

Dezavantajları:

- Büyük veri (big data) işleme konusunda yavaş kalabilir.
- Şema katı olduğundan esneklik düşüktür.
- Yatay ölçeklenme (çoklu sunucuda çalışma) sınırlıdır.
- JSON, XML gibi yarı-yapılandırılmış verilerle çalışmak zordur.

RDBMS & NoSQL Karşılaştırma

Kriter	RDBMS (İlişkisel Veritabanı)	NoSQL (Yapılandırılmamış Veritabanı)
Veri Yapısı	Tablo (satır/sütun)	JSON, belge, anahtar-değer, sütun, grafik
Şema (Schema)	Katı ve önceden tanımlı	Şemasız veya dinamik şema
Veri Bütünlüğü	Yüksek (ACID garantili)	Esnek (BASE yaklaşımı)
Sorgu Dili	SQL	Kendi API'leri, genellikle SQL dışı
İlişkiler	Tablo ilişkileri (JOIN destekli)	JOIN yok veya sınırlı
Ölçeklenebilirlik	Dikey (daha güçlü sunucuya ihtiyaç duyar)	Yatay (daha fazla sunucu eklenerek)

RDBMS & NoSQL Karşılaştırma

Kriter	RDBMS (İlişkisel Veritabanı)	NoSQL (Yapılandırılmamış Veritabanı)
Performans	Karmaşık sorgularda güçlü, ancak büyük veride yavaş	Büyük veri için yüksek hız, basit sorgularda çok hızlı
Uygun Olduğu Alanlar	Bankacılık, muhasebe, kurumsal uygulamalar	Sosyal medya, büyük veri, IoT, gerçek zamanlı uygulamalar
Veri Türleri	Yapılandırılmış veri	Yapılandırılmış, yarı-yapılandırılmış, yapılandırılmamış
ACID Desteği	Tam destekli (Atomicity, Consistency, Isolation, Durability)	Genellikle BASE (Basically Available, Soft state, Eventually consistent)
Örnek Sistemler	MySQL, PostgreSQL, Oracle, SQL Server	MongoDB, Cassandra, Redis, Neo4j