

NoSql ve MongoDB

Saygın Topatan

NoSql ve MongoDB

- ▶ **NoSql nedir**
 - ▶ Neden ihtiyaç duyuldu
 - ▶ Tipleri
- ▶ **MongoDb**
 - ▶ Kavramlar
 - ▶ Sharding
- ▶ **Şema Tasarımı**
- ▶ **NoSql'in geleceği**

NoSql Nedir?

Nedir

- ▶ 2009 başlarında ortaya çıkmış bir kavram
- ▶ İlişkisel Olmayan Veritabanı Yönetim Sistemi

Özellikleri

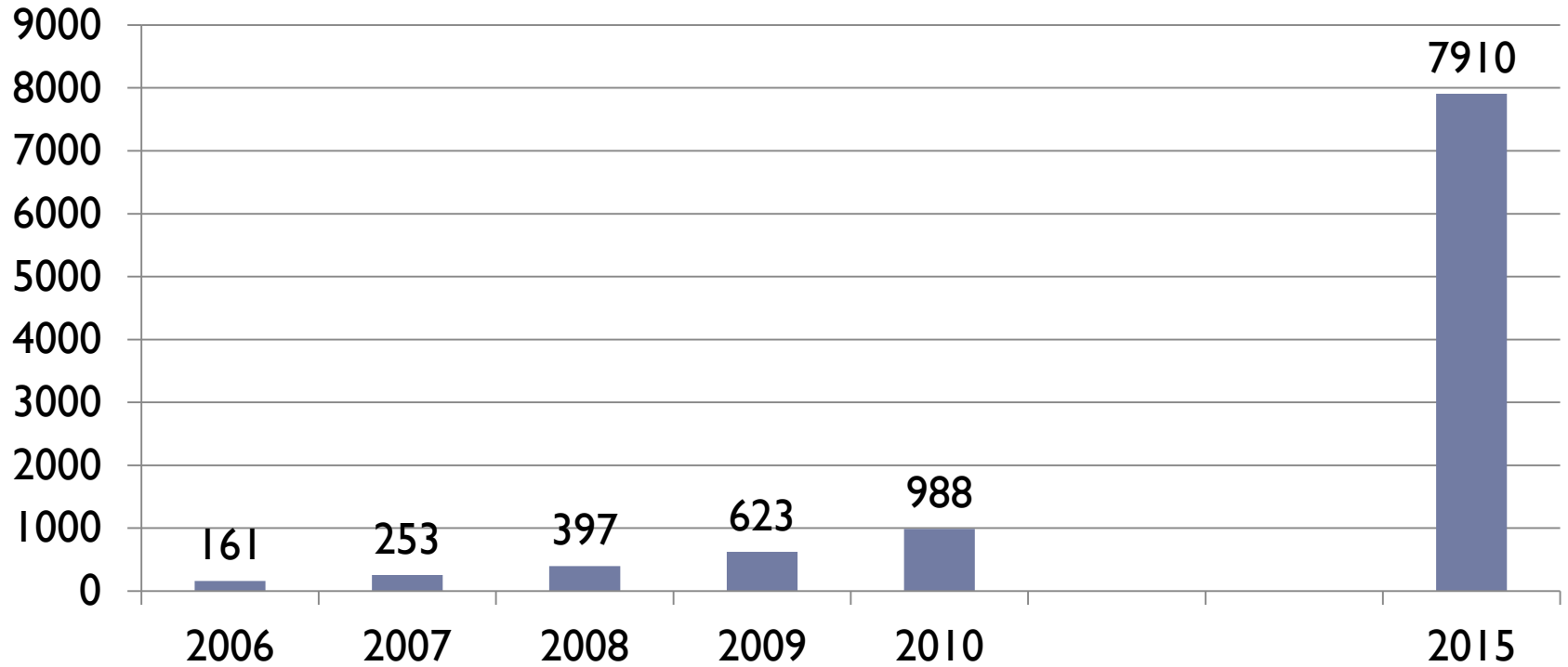
- ▶ Sabit tablo şemaları gerektirmeyebilir
- ▶ Join işlemlerinden kaçınır
- ▶ Yatay ölçeklenir!

No to Sql ?

Not
Only **SQL**

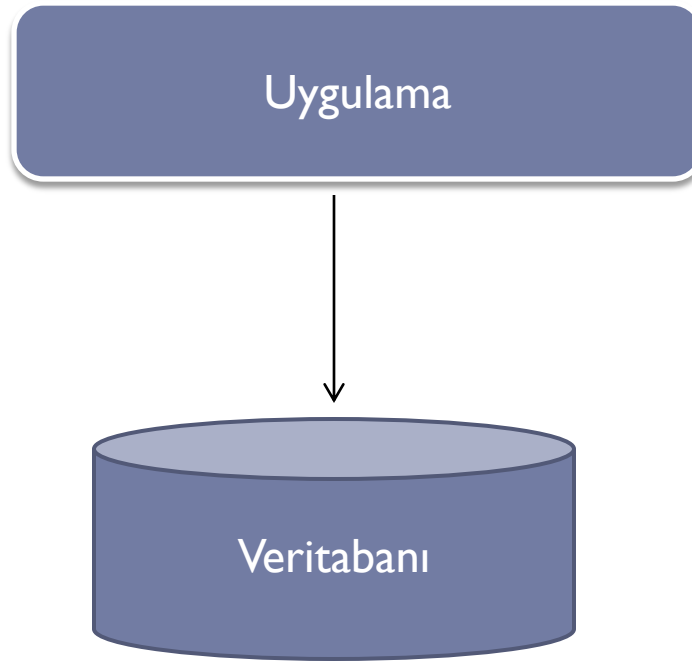
Neden NoSql? Veri Miktarı

Yıllara Göre Saklanan Veri Miktarı ExaByte (10^{18}) Kaynak: IDC 2011



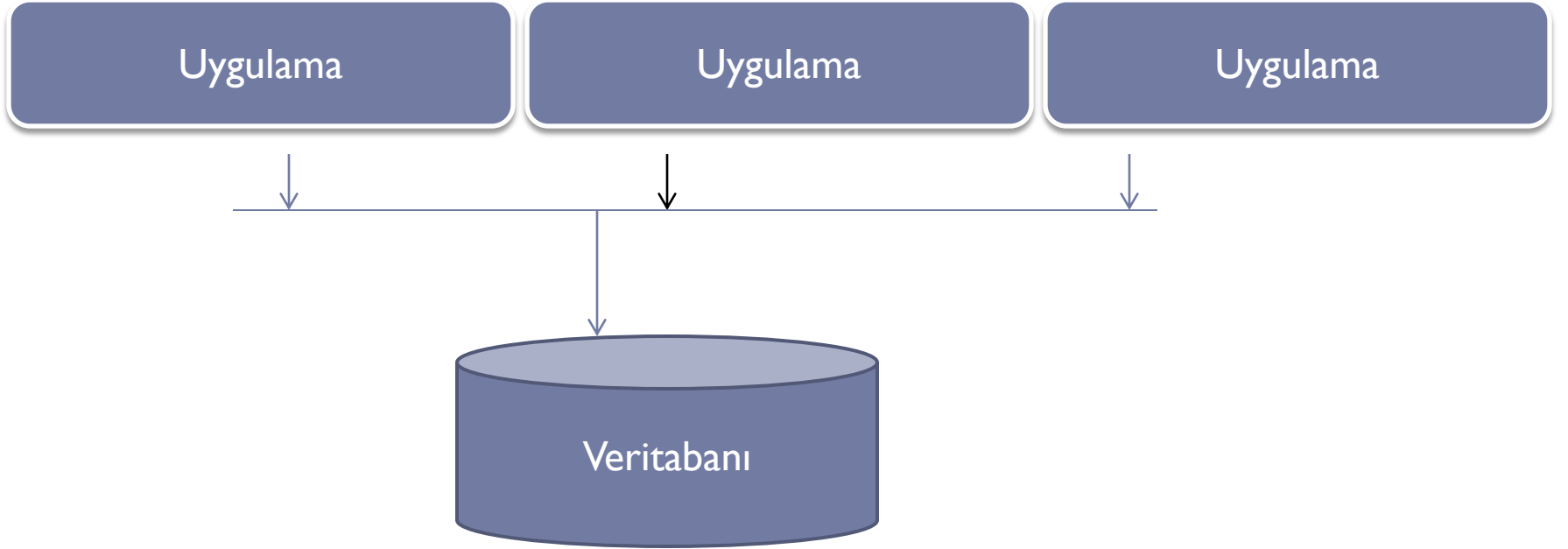
Neden NoSql? Uygulama Mimarileri

1980 ler: Mainframe Uygulamaları



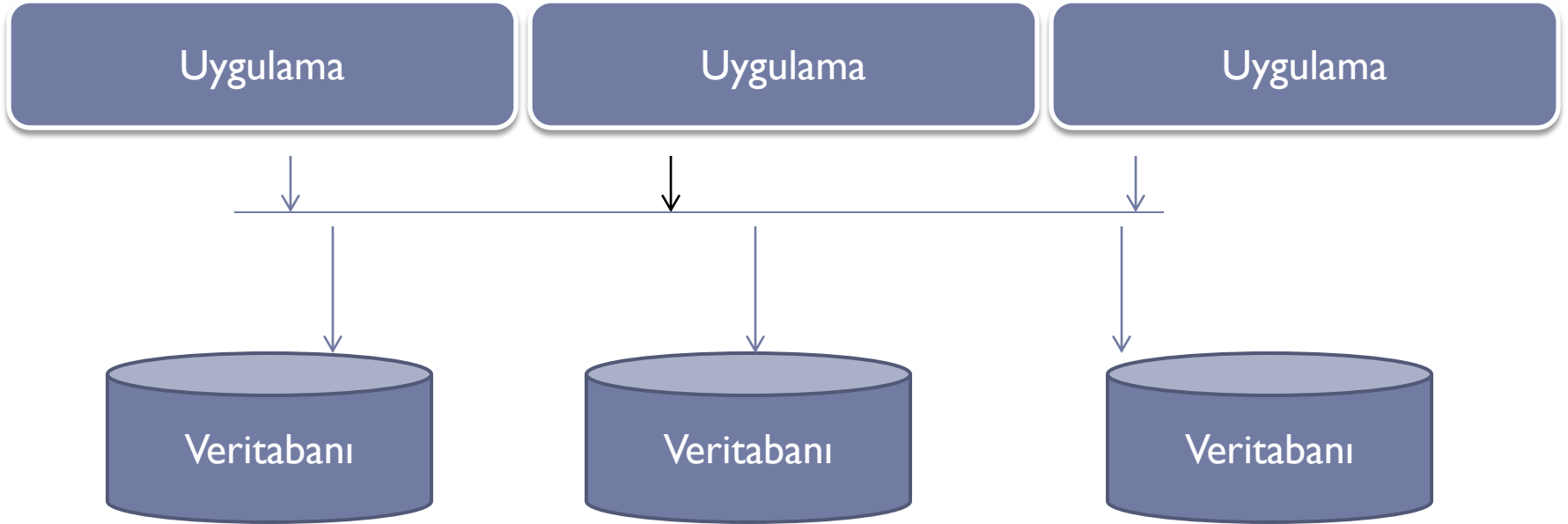
Neden NoSql? Uygulama Mimarileri

1990 ler: Entegrasyon merkezi olarak veritabanı



Neden NoSql? Uygulama Mimarileri

2000 ler: Artan saklama ve performans ihtiyaçları



Neden NoSql?

Google™

amazon.com®

Linked in®

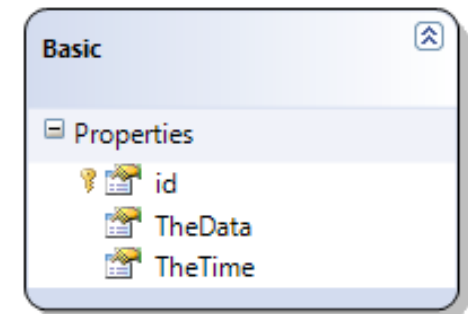
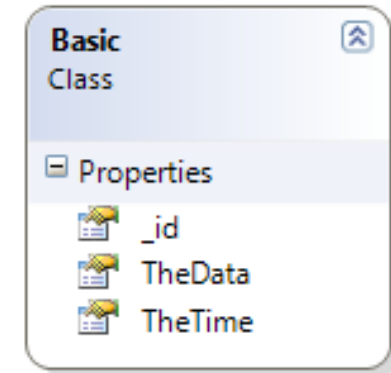
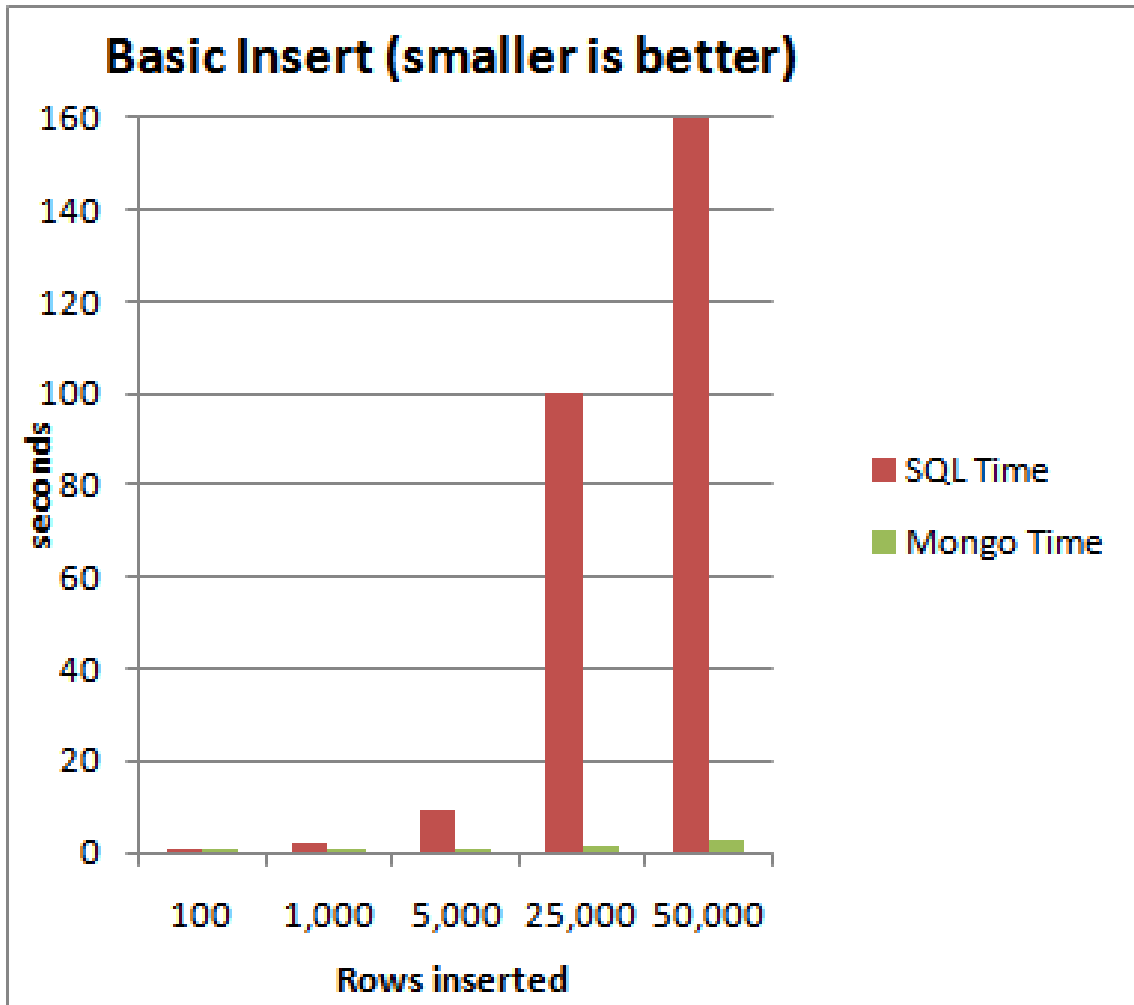
facebook®

twitter

Neden Ölçekleme?

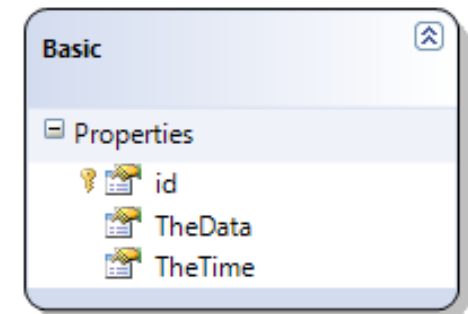
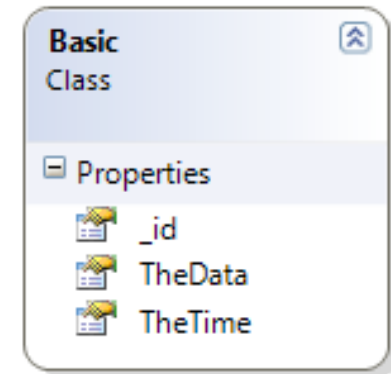
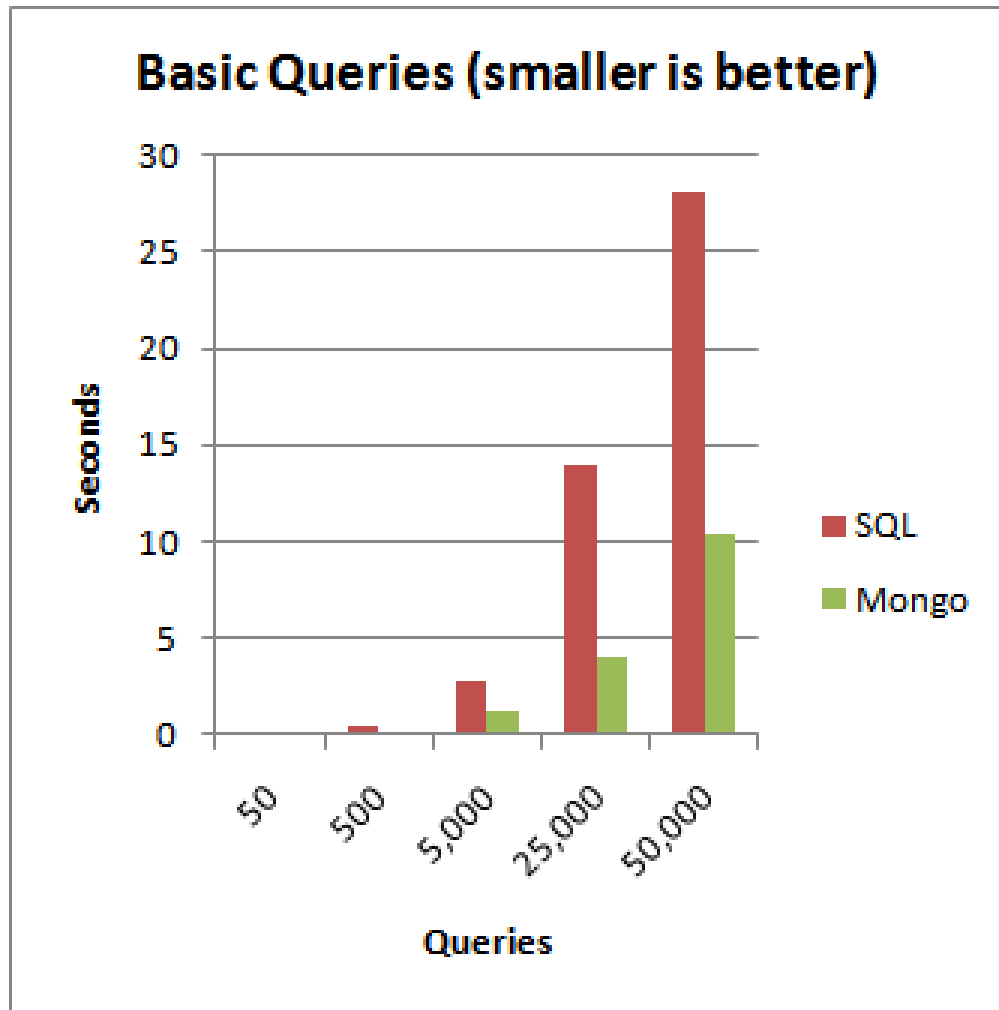
PowerEdge T110 II (Basic) – 8 GB, 3.1 Ghz Quad 4T	\$1,350.00
PowerEdge T110 II (Basic) – 32 GB, 3.4 Ghz Quad 8T	\$12,103.00
PowerEdge C2100 - 192 GB, 2 x 3 Ghz	\$19,960.00
IBM System x3850 X5 – 8 x 2.4 Ghz, 2048 GB	\$645,605.00
K Computer (süper bilgisayar) - 10 petaflops, 705,024 cores, 1,377 TB	\$10,000,000 yıllık işletim ücreti Setup ücreti ile ilgili bilgi yok

Neden NoSql? MongoDB vs. Sql Server 2008



MongoDB C# API
vs.
LINQ to SQL

Neden NoSql? MongoDB vs. Sql Server 2008

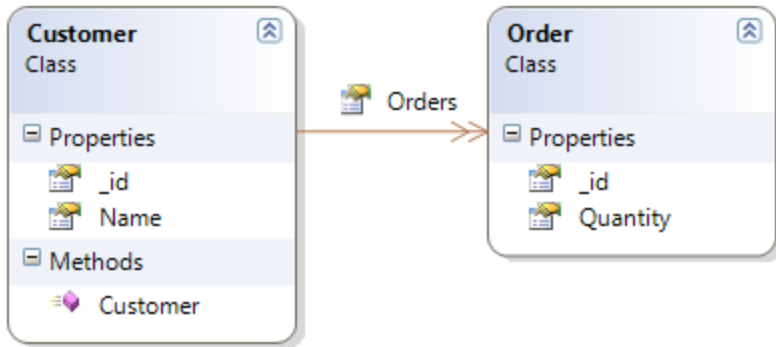


MongoDB C# API
vs.
LINQ to SQL

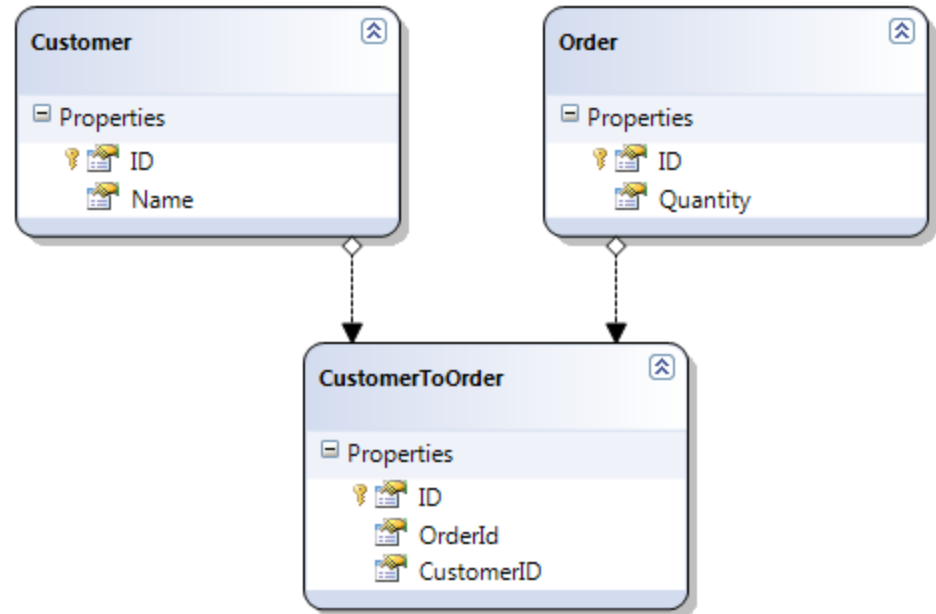
Neden NoSql? MongoDB vs. Sql Server 2008

Gerçek Dünya verileri

MongoDb



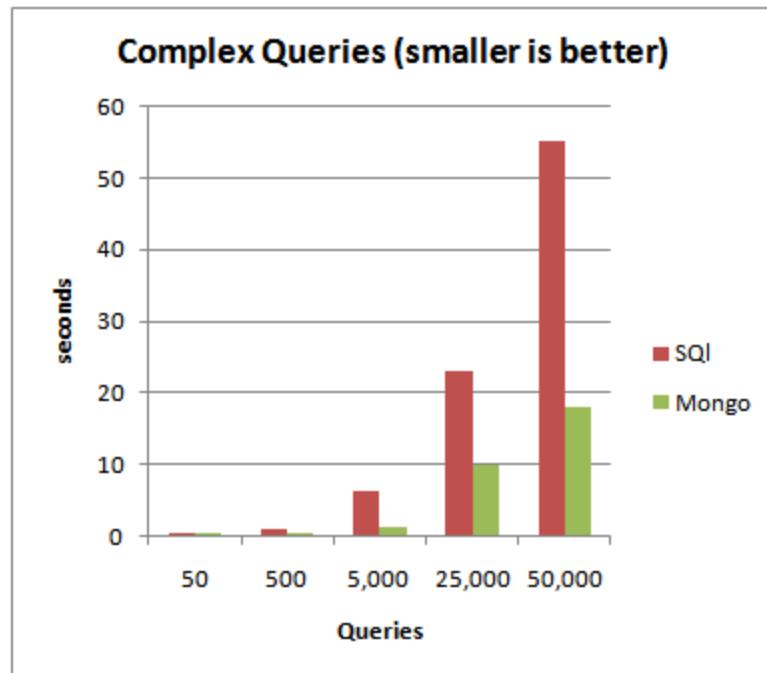
SQL Server



Neden NoSql? MongoDB vs. Sql Server 2008

Number of Parallel Clients 5 Time in seconds

	Total Rows	Rows / client	SQL Time	Mongo Time	Sql Ops/sec	Mongo Ops/sec
Complex / Real w query (2 joins in S embedded doc in MongoDB)	50	10	0.1	0.066	500	758
	500	100	1	0.19	500	2,632
	5,000	1,000	6.4	1.2	781	4,167
	25,000	5,000	23	9.9	1,087	2,525
	50,000	10,000	55	18	909	2,778
	1,000,000	200,000	960	398	1,042	2,513

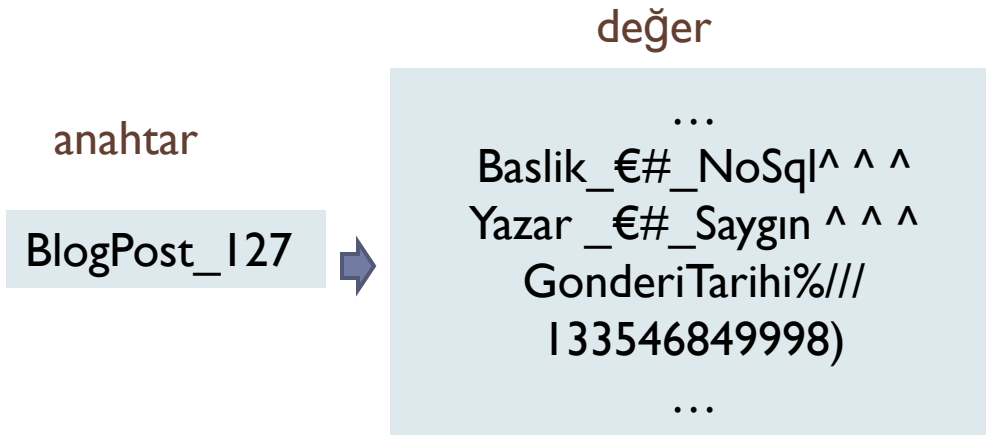


NoSql Tipleri

- ▶ **Anahtar-Değer Ambarı (Key-Value Stores)**
 - ▶ Dynamoite, Voldemort, Riak, Tokyo Cabinet
- ▶ **Doküman Ambarı (Document Store)**
 - ▶ MongoDB, CouchDB, RavenDB
- ▶ **Geniş Sütun Ambarı/Sütun Aileleri (Wide Column Store/Column Families, BigTable Clones)**
 - ▶ Apache Hbase, HyperTable, Cassandra
- ▶ **Graf Veritabanları (Graph Databases)**
 - ▶ Neo4j, AllegroGraph, InfoGrid

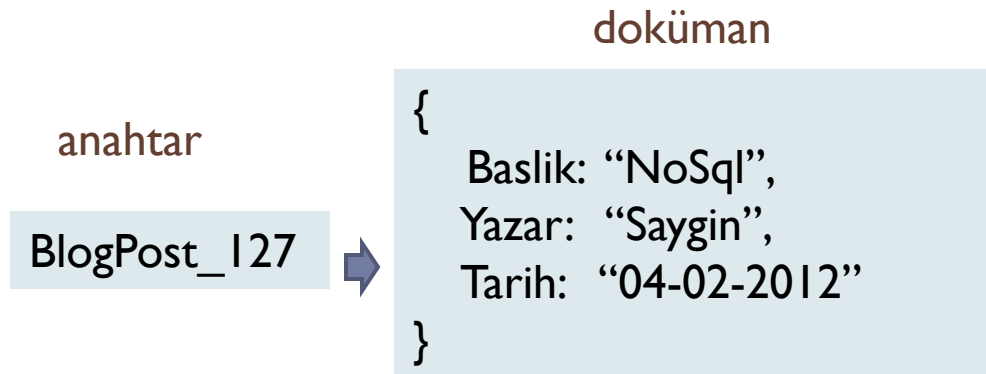
Anahtar Değer Ambarı

- ▶ Tek anahtar – tek değer
- ▶ Son derece hızlı
- ▶ Değer ikili (binary) olarak tutulur. Veritabanı değerini verisini anlamaz, anlamak istemez
- ▶ Nesneye erişim anahtar üzerinden olur



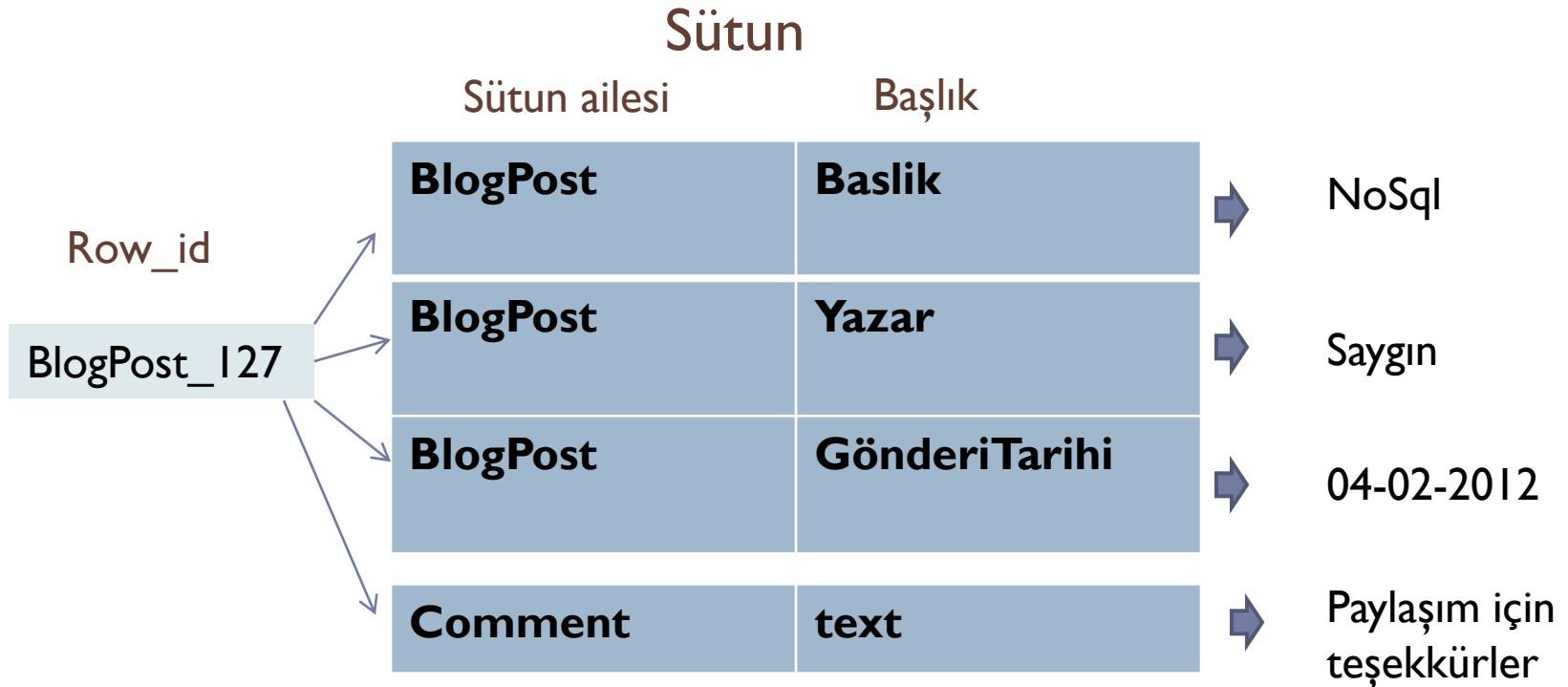
Doküman Ambarı (Document Store)

- ▶ Anahtar-değer ambarı, ama değer veritabanı tarafından anlaşılabilir
- ▶ Değer üzerinden sorgu yapmak mümkün



Geniş Sütun Ambarı/Sütun Aileleri (BigTable Clones)

- ▶ Google BigTable yayınından esinlenilmiştir
- ▶ Dağıtık, çok boyutlu, sıralanmış eşleme
- ▶ Her anahtar birden çok sütun ile eşleştirilir



Geniş Sütun Ambarı/Sütun Aileleri (BigTable Clones)

► Google BigTable 2006 verileri

Project name	Table size (TB)	Compression ratio	# Cells (billions)	# Column Families	# Locality Groups	% in memory	Latency-sensitive?
<i>Crawl</i>	800	11%	1000	16	8	0%	No
<i>Crawl</i>	50	33%	200	2	2	0%	No
<i>Google Analytics</i>	20	29%	10	1	1	0%	Yes
<i>Google Analytics</i>	200	14%	80	1	1	0%	Yes
<i>Google Base</i>	2	31%	10	29	3	15%	Yes
<i>Google Earth</i>	0.5	64%	8	7	2	33%	Yes
<i>Google Earth</i>	70	–	9	8	3	0%	No
<i>Orkut</i>	9	–	0.9	8	5	1%	Yes
<i>Personalized Search</i>	4	47%	6	93	11	5%	Yes

Graf Veritabanları

- ▶ Veri ilişkilerine odaklanmış graf yapısıdır.
- ▶ D ğ mler(nodes) ve bunları bađlayan kenarlar(edges)
- ▶ Her ikisinde de anahtar-deđer ikilileri bulunur

Yusuf

Gizem

Emre

Aslı

Graf Veritabanları

- ▶ Veri ilişkilerine odaklanmış graf yapısıdır.
- ▶ D ğ mler(nodes) ve bunları baėlayan kenarlar(edges)
- ▶ Her ikisinde de anahtar-deėer ikilileri bulunur

Yaş: 23
Okul: "İTÜ"

Yusuf

Yaş: 22
Okul: "İTÜ"

Gizem

Yaş: 22
Okul: "YTÜ"

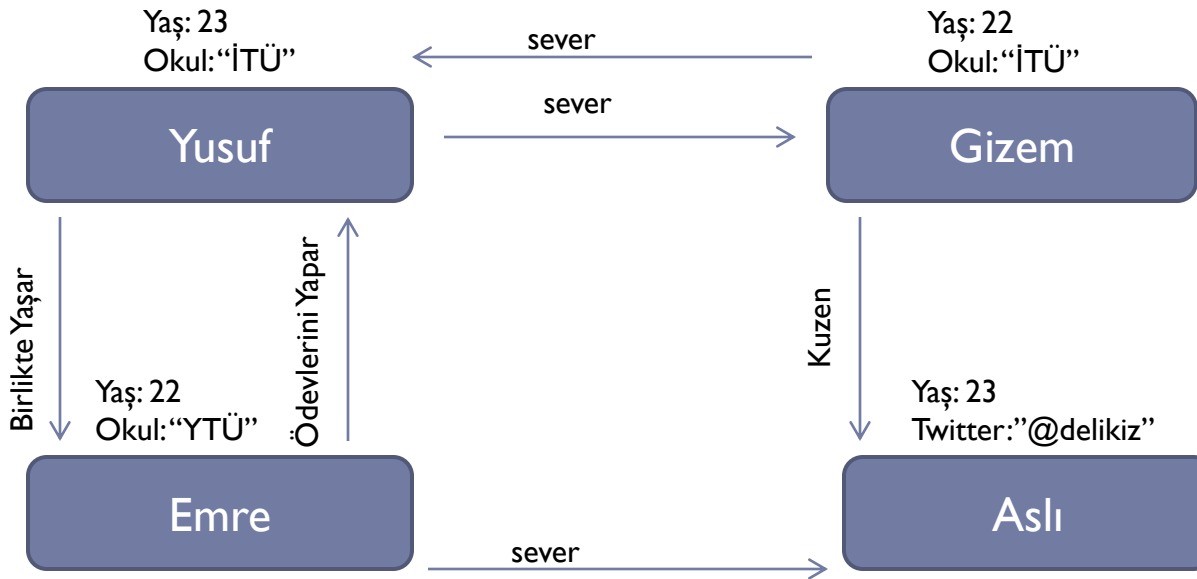
Emre

Yaş: 23
Twitter: "@delikiz"

Aslı

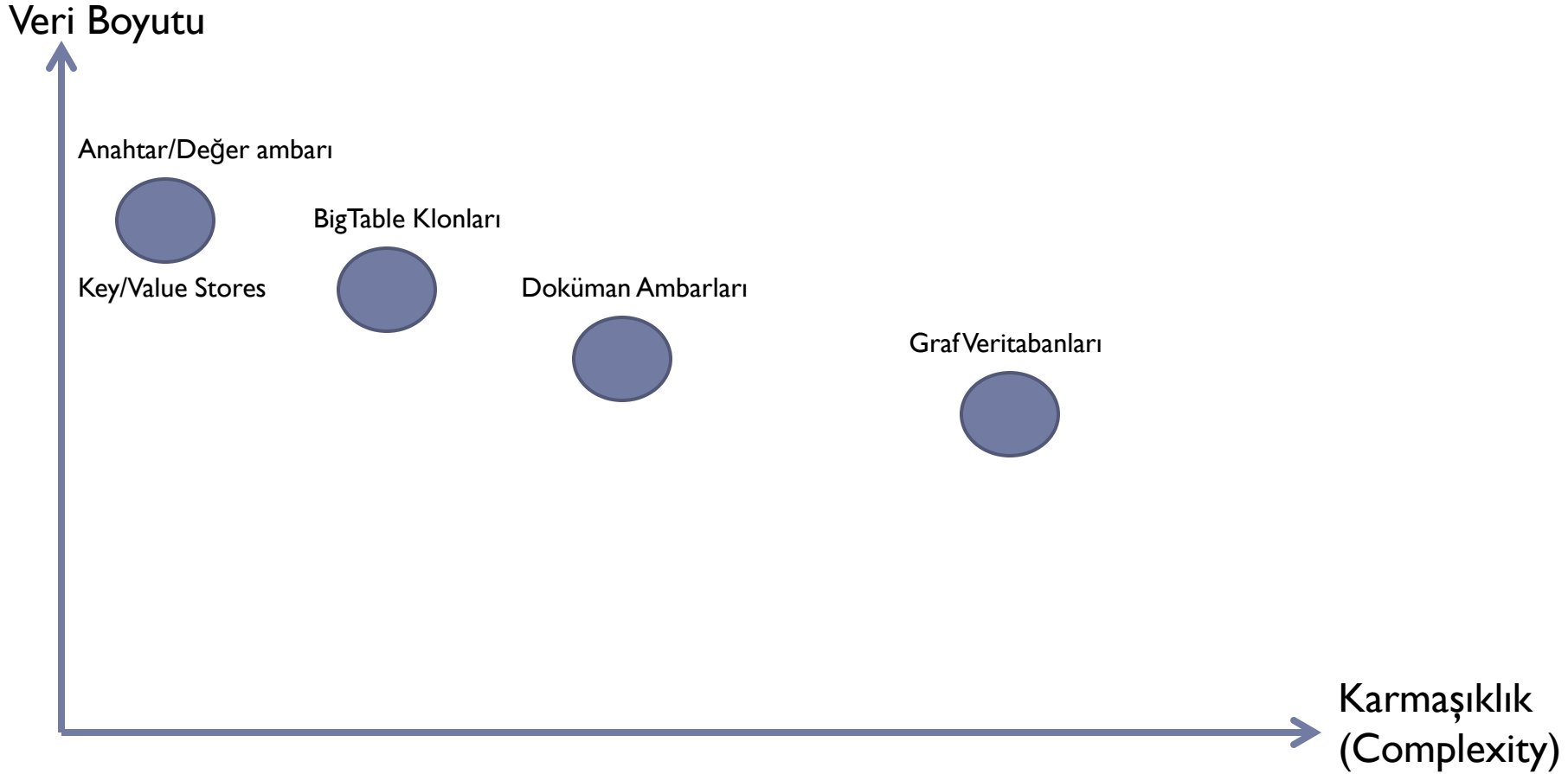
Graf Veritabanları

- ▶ Veri ilişkilerine odaklanmış graf yapısıdır.
- ▶ D ğ mler(nodes) ve bunları baėlayan kenarlar(edges)
- ▶ Her ikisinde de anahtar-deėer ikilileri bulunur



NoSql Tipleri

► Boyut-Karmaşıklık karşılaştırması





mongoDB

- ▶ **Doküman Tabanlı veritabanı**
 - ▶ JSON türevi (BSON) bir format kullanır
- ▶ **Şemasız**
- ▶ **Performans**
 - ▶ C++ da yazılmış
 - ▶ Index desteği
- ▶ **Ölçeklenebilir**
 - ▶ Replication
 - ▶ Auto-Sharding
- ▶ **Ticari destekli (10gen)**
 - ▶ Bolca doküman



mongoDB

The image displays a collection of logos for various programming languages and frameworks, arranged in a grid-like fashion. The logos include:

- C**: The cover of the book "The C Programming Language" by Brian Kernighan and Dennis Ritchie.
- C++**: The cover of the book "C++: The Basics" by Bjarne Stroustrup.
- F#**: The cover of the book "Expert F#" by Don Syme.
- C#**: The logo for the C# programming language.
- PHP**: The logo for the PHP programming language.
- Python**: The logo for the Python programming language.
- nodeJS**: The logo for Node.js.
- GO**: The logo for the Go programming language, featuring the word "GO" in a stylized font with two horizontal lines above it.
- ERLANG**: The logo for the Erlang programming language, featuring a red and white stylized "E" and the word "ERLANG" below it.
- CF**: The logo for the ColdFusion programming language, featuring the letters "CF" in white on a blue square background.
- Scala**: The logo for the Scala programming language, featuring a red and white stylized "S" and the word "Scala" in black.
- Java**: The logo for the Java programming language, featuring a stylized orange and blue flame above the word "Java" in orange.
- HASKELL**: The logo for the Haskell programming language, featuring the word "HASKELL" in green.

<http://www.mongodb.org/display/DOCS/Drivers>



mongoDB

- ▶ Database == Database
- ▶ Table == Collection
- ▶ Index == Index
- ▶ Row == BSON Document
- ▶ Column == BSON Field

mongoDB

CREATE TABLE USERS (a Number, b Number) Table
db.createCollection("mycoll")

ALTER TABLE users ADD ...

mongoDB

SELECT * FROM users

db.users.find()

SELECT a, b FROM users

db.users.find({ }, {a:1, b:1})

SELECT * FROM users WHERE age>33

db.users.find({age:{\$gt:33}})

SELECT * FROM users WHERE age=33 ORDER BY name

db.users.find({age:33}).sort({name:1})

mongoDB

```
SELECT DISTINCT last_name FROM users  
db.users.distinct('last_name')
```

```
SELECT COUNT(*) FROM users  
db.users.count()
```

```
SELECT COUNT(AGE) from users  
db.users.find({'$exists': true}).count()
```

mongodb Indexleri

// Artan

```
db.users.ensureIndex({name:1})
```

// Eşsiz

```
db.users.ensureIndex({name:1}, {unique:true})
```

//Sessiz, Bloklamayan

```
db.users.ensureIndex({name:1}, {background:true})
```

//Bileşik

```
db.users.ensureIndex({name:1}, {age:-1})
```

mongoDB Auto Sharding

- ▶ Verinin her parçası ayrı bir sunucu (shard) tarafından yönetilir
- ▶ Her bir sunucu da verinin bir alt kümesi üzerinde tam kontrole sahiptir
- ▶ Veritabanı işlemleri mümkünse bir shard üzerinde, değilse birden çok sunucu üzerinde yapılır
- ▶ Sistem yükü değiştikçe, verinin sunuculara dağıtılması otomatik olarak dengelenir

Auto Sharding Mimari Görünümü

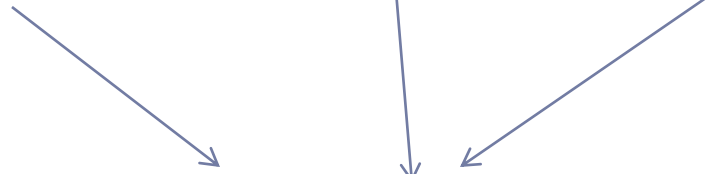
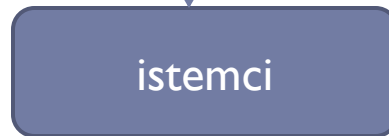
$0 \leq \text{userId} < 2000$



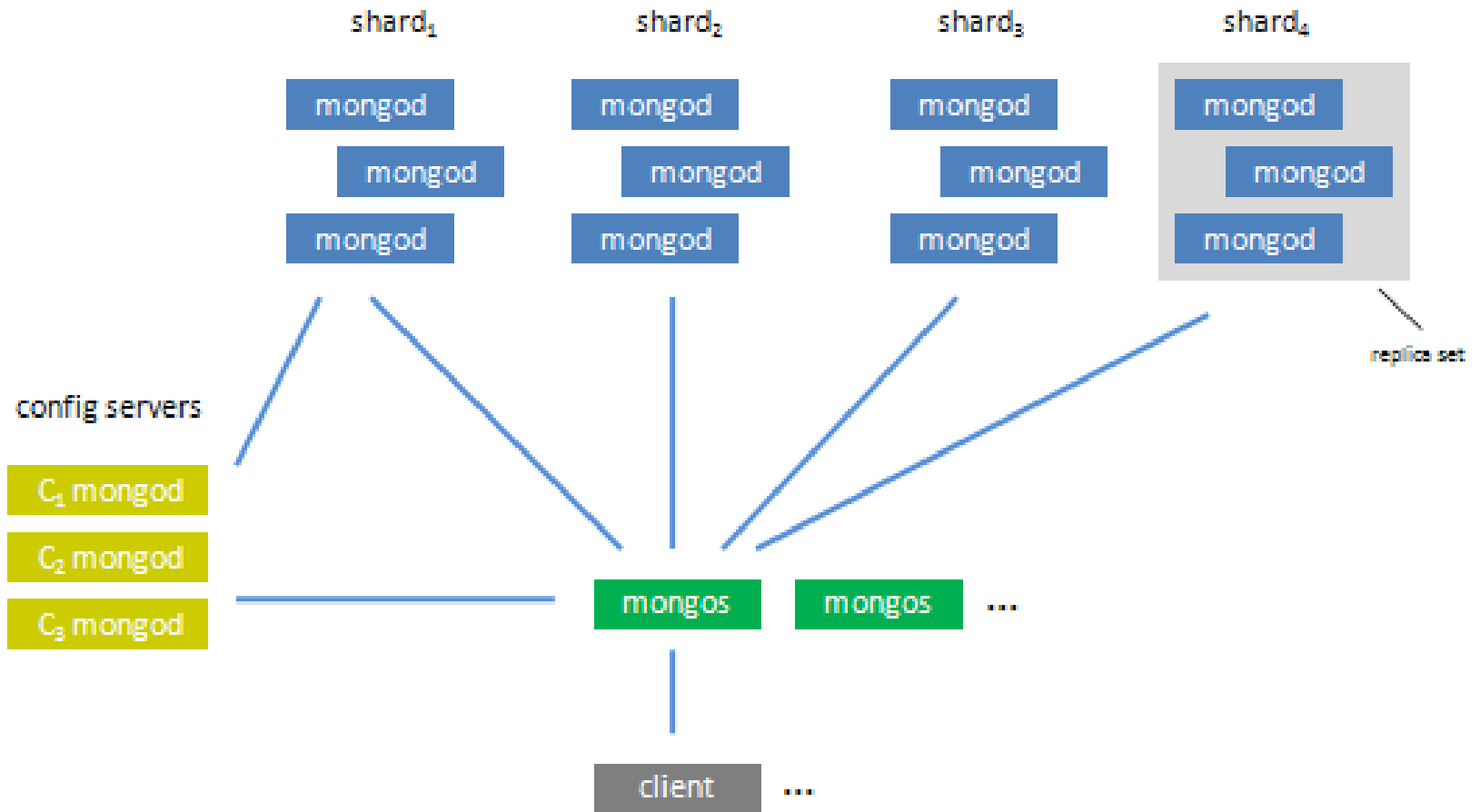
$2000 \leq \text{userId} < 5000$



$5000 \leq \text{userId} < 9000$



Auto Sharding Mimari Görünümü



Şema Tasarımı

- ▶ **Erişim?**
 - ▶ OLAP – OLTP
 - ▶ Güncelleme tipleri
 - ▶ Sorgu Tipleri
- ▶ **Dikkat!**
 - ▶ Join yok
 - ▶ Transaction yok
- ▶ **Şema performansı etkileyen en önemli faktör**
 - ▶ RDBMS e göre daha fazla tasarım seçeneği var
 - ▶ Embedding, indexler, shard anahtarları

Şema Tasarımı

► Şemasız

Id	Tip	Alan	Yaricap	Kenar	Genislik	Uzunluk
1	Çember	3.14	1			
2	Kare	4		2		
3	Dikdörtgen	10			2	5

db.sekil.find()

{id:"1", tip:"çember", alan:3.14, yaricap:1}

{id:"2", tip:"kare", alan:4, kenar:1}

{id:"3", tip:"dikdortgen", alan:10, genislik:2, uzunluk:5}

Blog Post - Gömülü

```
{
  id:"post/saygin/2012-02-04/",
  yazar:"saygin",
  baslik:"NoSql ve MongoDB",
  etiketler:["mongodb","nosql"],
  yorumlar:[
    {
      yazar:"Yusuf",
      baslik:"Harika !",
      tarih:"2012-02-04"
    }
  ]
}
```

Okuma performansı için çok iyi
Tüm nesneyi tek seferde yüklüyor
Veritabanına tek git-gel yapıyor
Sürekli yazma ihtiyacı varsa, yazma yavaş olabilir

Blog Post – Gömülü Değil

Blog.posts

```
{  
  id:"post/saygin/2012-02-04/",  
  yazar:"saygin",  
  baslik:"NoSql ve MongoDB",  
  etiketler:["mongodb","nosql"]  
}
```

Blog.yorumlar

```
{  
  post:"post/saygin/2012-02-04/",  
  yazar:"Yusuf",  
  baslik:"Harika !",  
  tarih:"2012-02-04"  
}
```

Blog Post - Hibrid

Blog.yorumlar

```
{
  id: "post/saygin/2012-02-04/1----1",
  yorumlar:[
    {
      yazar:"Yusuf",
      baslik:"Harika !",
      tarih:"2012-02-04"
    },
    {
      yazar:"Hasan",
      baslik:"Güzel Paylaşım",
      tarih:"2012-02-04"
    },
  ]
}
```

İndeksler

- ▶ Ortak sorgulara göre indeksler oluşturulmalı
- ▶ Duplike indeksler olmamalı : (A,B) varsa (A) gereksiz
- ▶ Gereksiz indeksler yazma-güncelleme performansını etkiler

Shard Anahtarı Seçmek

- ▶ Shard anahtarı verinin nasıl dağıtılacağını belirler
- ▶ Değiştirmesi zordur
- ▶ En önemli performans kararı (Sharding)

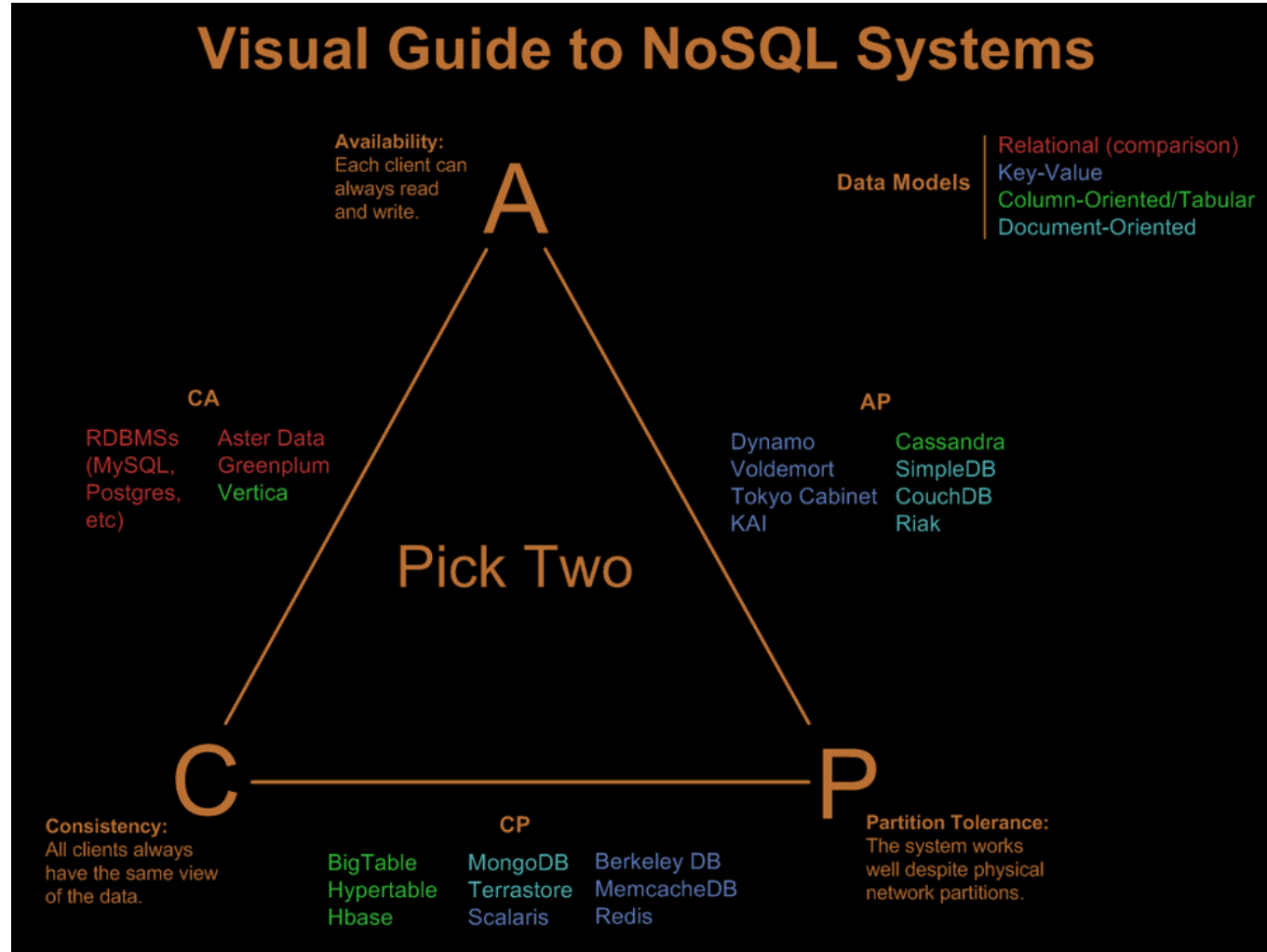
NoSql in geleceđi

▶ RDBMS

- ▶ Veri bütünlüğü
- ▶ Normalizasyon
- ▶ Mevcut sistemler, üreticiler (Oracle, microsoft..)
- ▶ Veritabanı araçları (Raporlama, analiz, 3rd party)

- ▶ Ölçekleme problemleri

Doğru Aracı Seçmek



Teşekkürler

Saygın Topatan

Saygin@sirkethaberleri.com

@saygin

