



NoSQL Design Workshop





Facebook somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help

somkiat.cc
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ...

+ Add a Button



NoSQL



Database Ranking

422 systems in ranking, September 2023

Rank			DBMS	Database Model	Score		
Sep 2023	Aug 2023	Sep 2022			Sep 2023	Aug 2023	Sep 2022
1.	1.	1.	Oracle 	Relational, Multi-model 	1240.88	-1.22	+2.62
2.	2.	2.	MySQL 	Relational, Multi-model 	1111.49	-18.97	-100.98
3.	3.	3.	Microsoft SQL Server 	Relational, Multi-model 	902.22	-18.60	-24.08
4.	4.	4.	PostgreSQL 	Relational, Multi-model 	620.75	+0.37	+0.29
5.	5.	5.	MongoDB 	Document, Multi-model 	439.42	+4.93	-50.21
6.	6.	6.	Redis 	Key-value, Multi-model 	163.68	+0.72	-17.79
7.	7.	7.	Elasticsearch	Search engine, Multi-model 	138.98	-0.94	-12.46
8.	8.	8.	IBM Db2	Relational, Multi-model 	136.72	-2.52	-14.67
9.	↑ 10.	↑ 10.	SQLite 	Relational	129.20	-0.72	-9.62
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	128.56	-1.78	-11.47
11.	11.	↑ 13.	Snowflake 	Relational	120.89	+0.27	+17.39
12.	12.	↓ 11.	Cassandra 	Wide column, Multi-model 	110.06	+2.67	-9.06
13.	13.	↓ 12.	MariaDB 	Relational, Multi-model 	100.45	+1.80	-9.70
14.	14.	14.	Splunk	Search engine	91.40	+2.42	-2.65
15.	↑ 16.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-model 	82.73	+3.22	-1.69
16.	↓ 15.	↓ 15.	Amazon DynamoDB 	Multi-model 	80.91	-2.64	-6.51

<https://db-engines.com/en/ranking>



NoSQL

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

Relational Database

422 systems in ranking, September 2023

Rank			DBMS	Database Model	Score		
Sep 2023	Aug 2023	Sep 2022			Sep 2023	Aug 2023	Sep 2022
1.	1.	1.	Oracle +	Relational, Multi-model i	1240.88	-1.22	+2.62
2.	2.	2.	MySQL +	Relational, Multi-model i	1111.49	-18.97	-100.98
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	902.22	-18.60	-24.08
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	620.75	+0.37	+0.29
5.	5.	5.	MongoDB +	Document, Multi-model i	439.42	+4.93	-50.21
6.	6.	6.	Redis +	Key-value, Multi-model i	163.68	+0.72	-17.79
7.	7.	7.	Elasticsearch	Search engine, Multi-model i	138.98	-0.94	-12.46
8.	8.	8.	IBM Db2	Relational, Multi-model i	136.72	-2.52	-14.67
9.	↑ 10.	↑ 10.	SQLite +	Relational	129.20	-0.72	-9.62
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	128.56	-1.78	-11.47
11.	11.	↑ 13.	Snowflake +	Relational	120.89	+0.27	+17.39
12.	12.	↓ 11.	Cassandra +	Wide column, Multi-model i	110.06	+2.67	-9.06
13.	13.	↓ 12.	MariaDB +	Relational, Multi-model i	100.45	+1.80	-9.70
14.	14.	14.	Splunk	Search engine	91.40	+2.42	-2.65
15.	↑ 16.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-model i	82.73	+3.22	-1.69
16.	↓ 15.	↓ 15.	Amazon DynamoDB +	Multi-model i	80.91	-2.64	-6.51

<https://db-engines.com/en/ranking>



NoSQL

© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

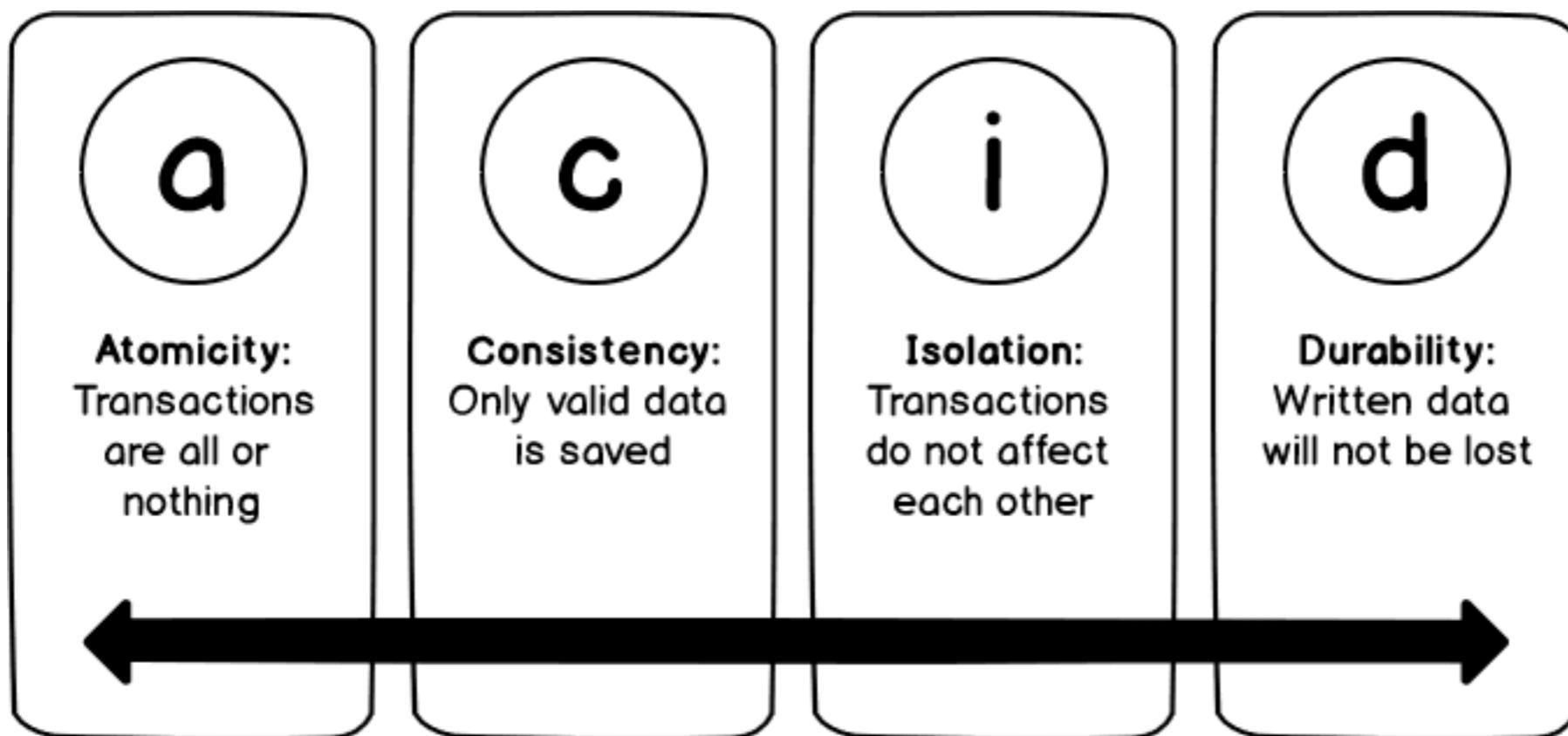
Relational Database



```
SELECT *
FROM SOME_TABLE
WHERE 1=1
HAVING ...
GROUP BY ...
```



A.C.I.D



Other Database Models

422 systems in ranking, September 2023

Rank			DBMS	Database Model	Score		
Sep 2023	Aug 2023	Sep 2022			Sep 2023	Aug 2023	Sep 2022
1.	1.	1.	Oracle +	Relational, Multi-model i	1240.88	-1.22	+2.62
2.	2.	2.	MySQL +	Relational, Multi-model i	1111.49	-18.97	-100.98
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	902.22	-18.60	-24.08
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	620.75	+0.37	+0.29
5.	5.	5.	MongoDB +	Document, Multi-model i	439.42	+4.93	-50.21
6.	6.	6.	Redis +	Key-value, Multi-model i	163.68	+0.72	-17.79
7.	7.	7.	Elasticsearch	Search engine, Multi-model i	138.98	-0.94	-12.46
8.	8.	8.	IBM Db2	Relational, Multi-model i	136.72	-2.52	-14.67
9.	↑ 10.	↑ 10.	SQLite +	Relational	129.20	-0.72	-9.62
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	128.56	-1.78	-11.47
11.	11.	↑ 13.	Snowflake +	Relational	120.89	+0.27	+17.39
12.	12.	↓ 11.	Cassandra +	Wide column, Multi-model i	110.06	+2.67	-9.06
13.	13.	↓ 12.	MariaDB +	Relational, Multi-model i	100.45	+1.80	-9.70
14.	14.	14.	Splunk	Search engine	91.40	+2.42	-2.65
15.	↑ 16.	↑ 16.	Microsoft Azure SQL Database	Relational, Multi-model i	82.73	+3.22	-1.69
16.	↓ 15.	↓ 15.	Amazon DynamoDB +	Multi-model i	80.91	-2.64	-6.51

<https://db-engines.com/en/ranking>



NoSQL

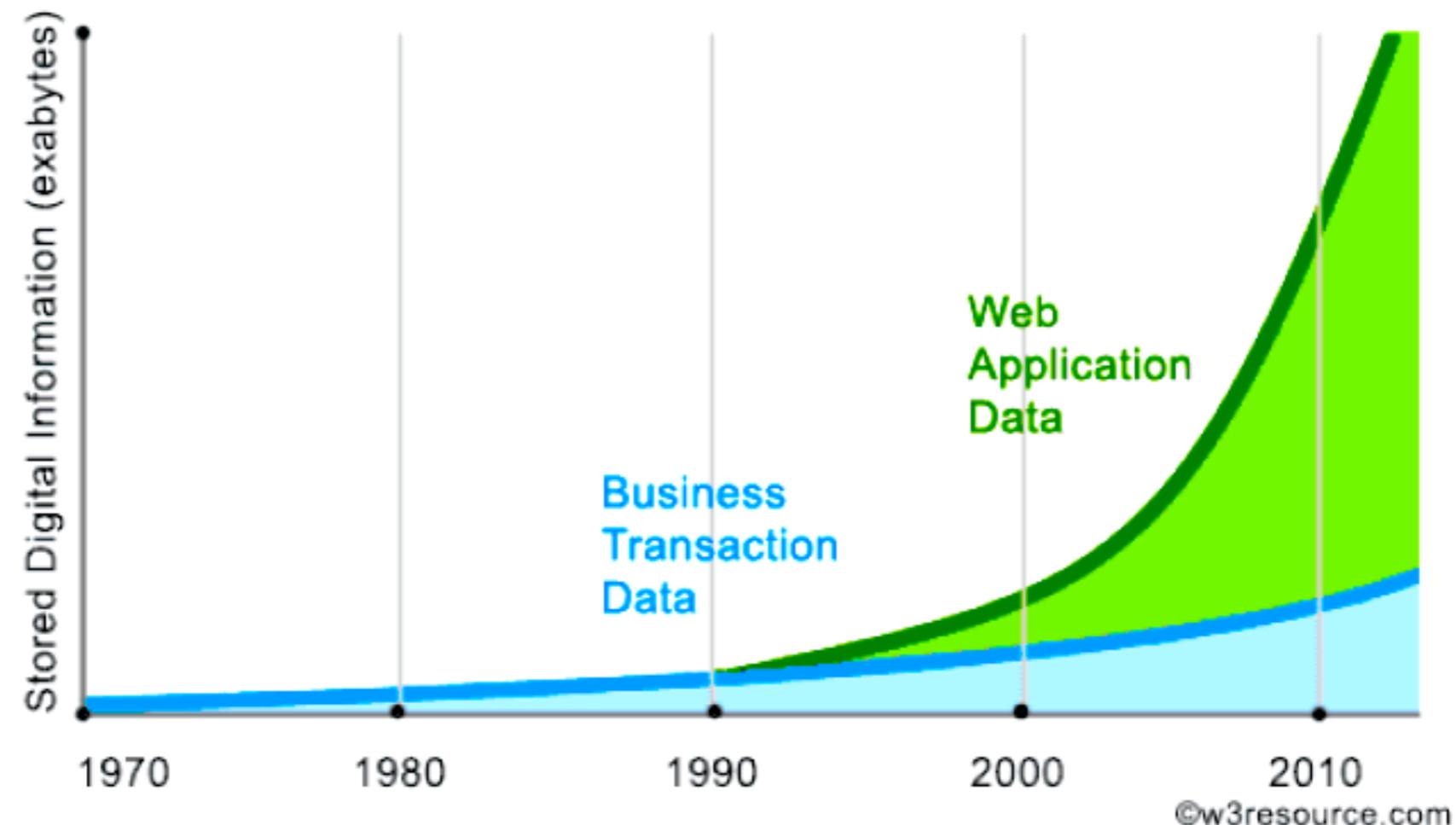
© 2017 - 2018 Siam Chamnkit Company Limited. All rights reserved.

Why ?



Growth of Data !!

Web Applications Driving Data Growth

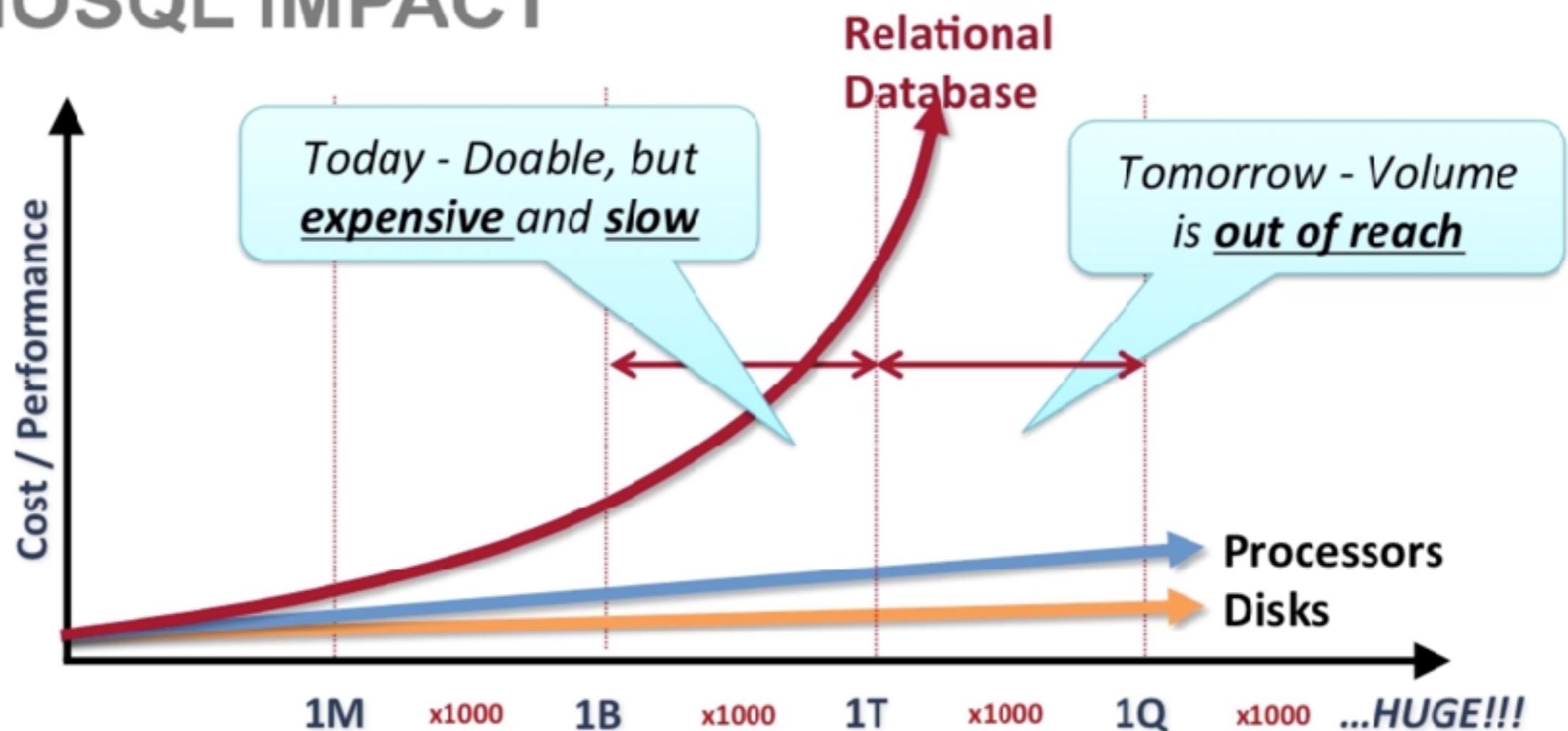


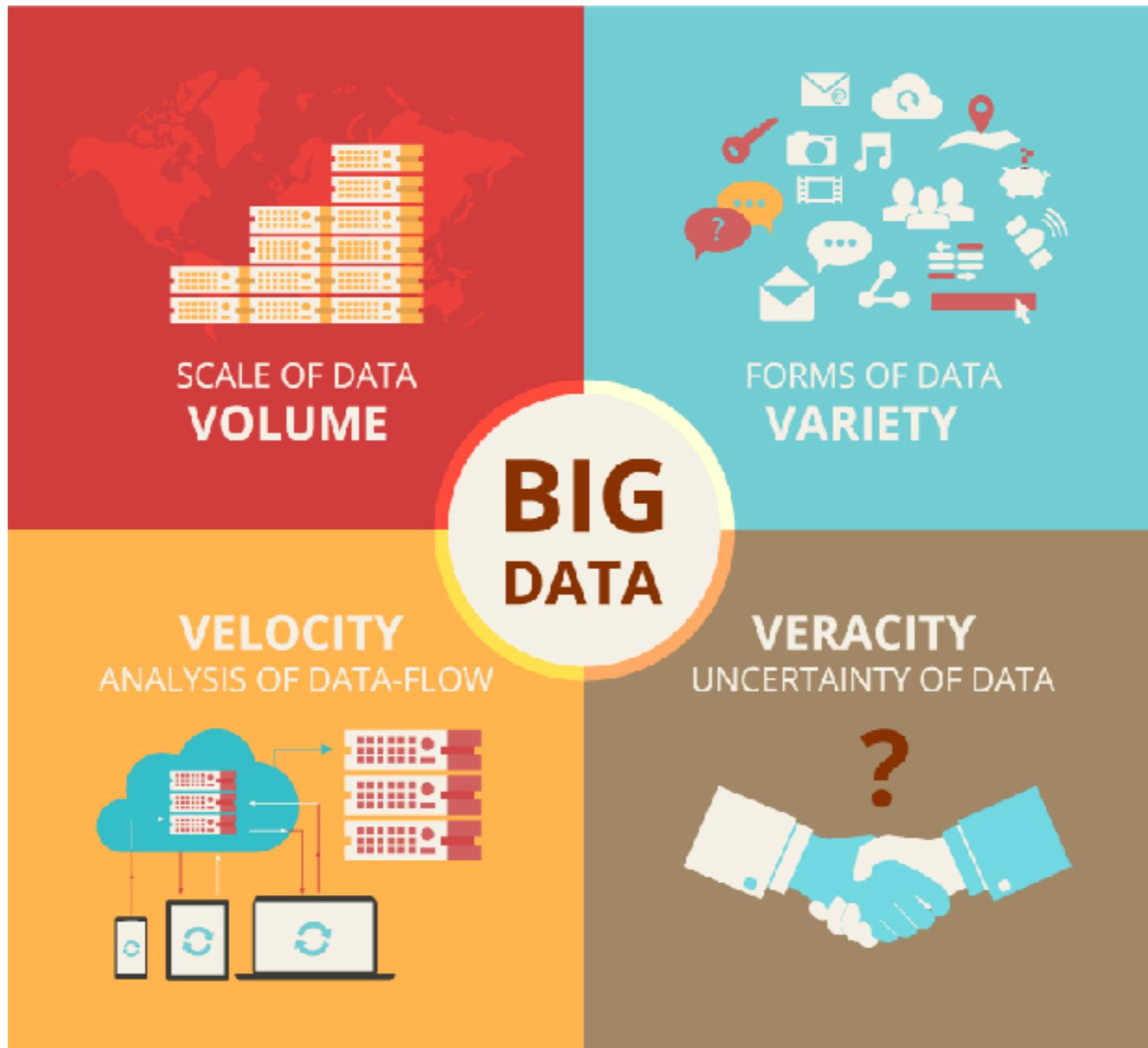
©w3resource.com



Impact with database !!

NOSQL IMPACT

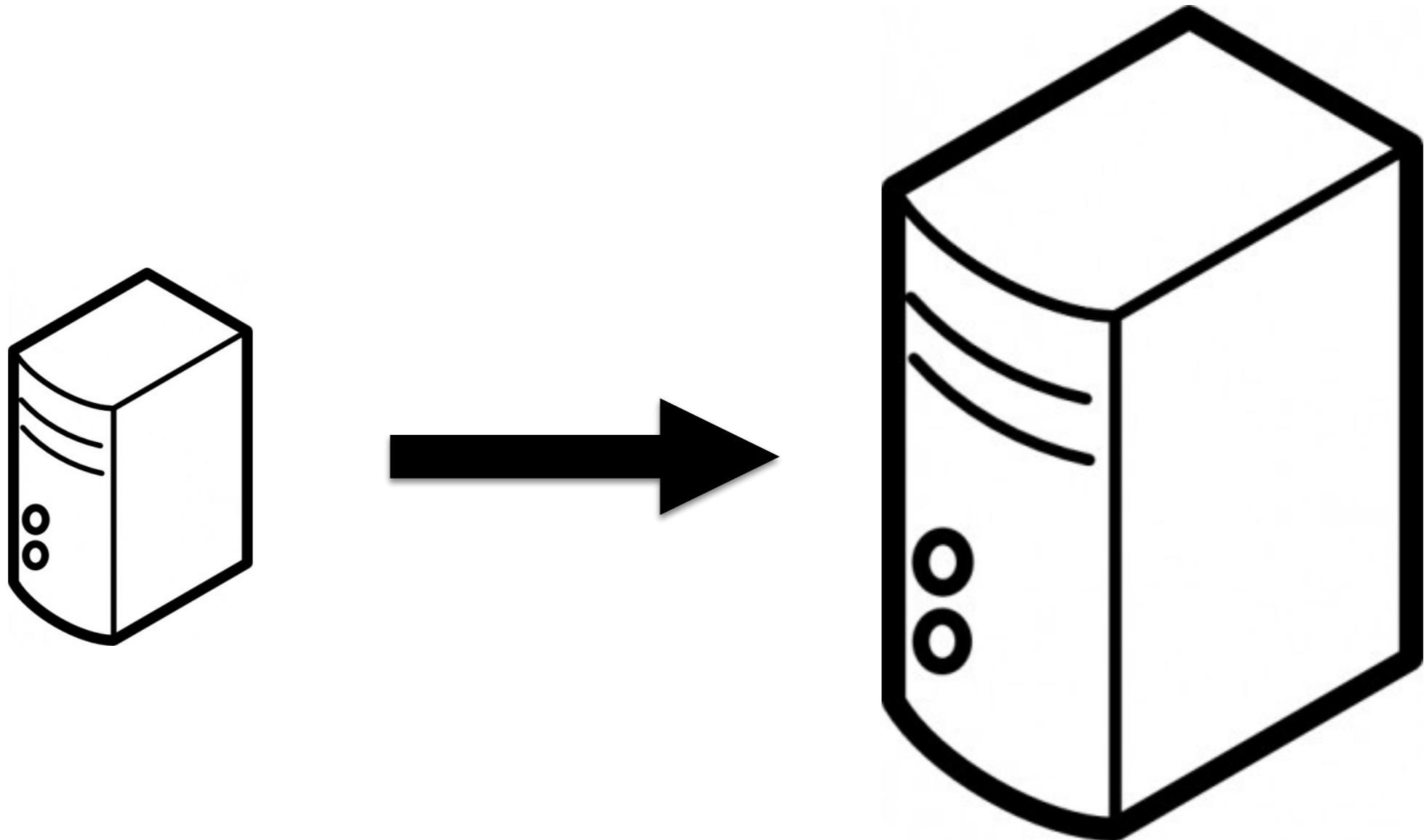




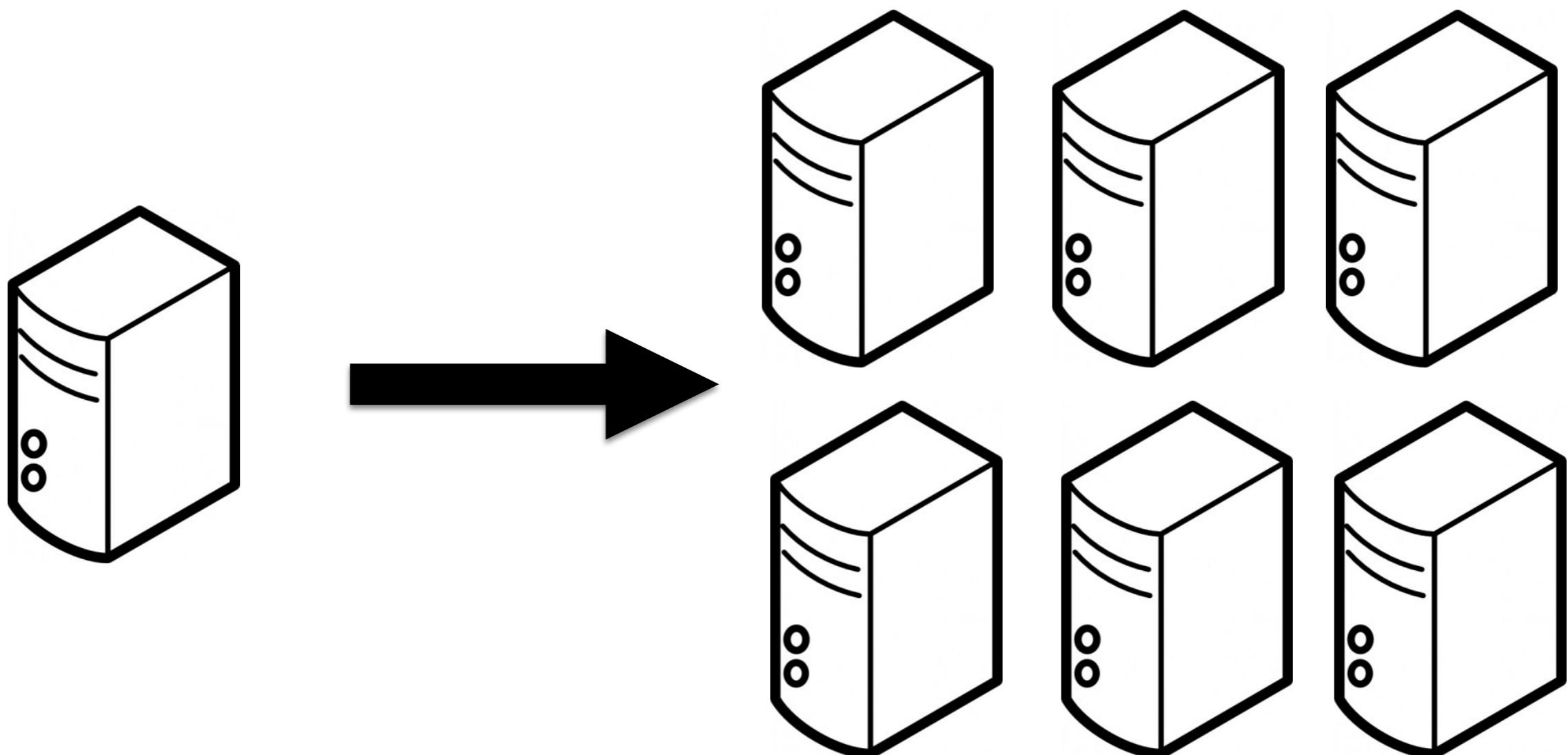
How to scale ?



Scale Up



Scale Out

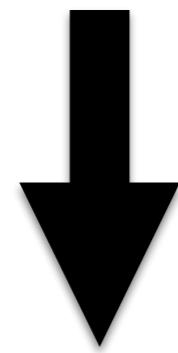


Performance vs Cost ?

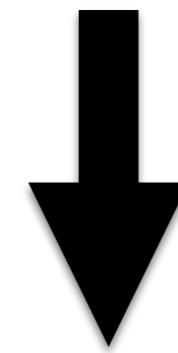


Need new way ?





Bigtable



DynamoDB



“NoSQL”





⌚ This event has ended

NOSQL meetup

Last.fm

Thursday, June 11, 2009 from 10:00 AM to 5:00 PM (PDT)
San Francisco, CA

Ticket Information

TYPE	REMAINING	END	QUANTITY
Free ticket	Sold Out	Ended	Free Sold Out

Share NOSQL meetup



Share



Tweet



Like

Be the first of your friends to like this.



What is NoSQL ?

No SQL
Not Only SQL
Non-relational



Properties of NoSQL

Non-relational (no join)

Open source

Cluster friendly (distributed system)

Schema-less / Dynamic schema

For new web app/application



Non-relational ?



Cassandra

APACHE
HBASE

Column Family



HYPERTABLE INC



Graph



Couchbase



Document



Apache



Project Voldemort
A distributed database.

Key-Value



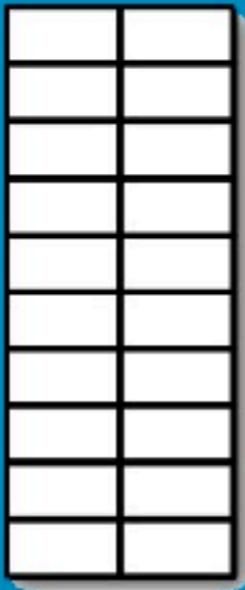
redis



NoSQL

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

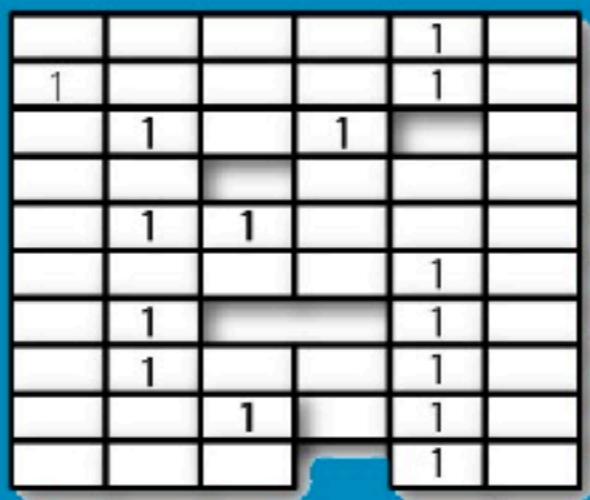
Key-Value



Graph DB

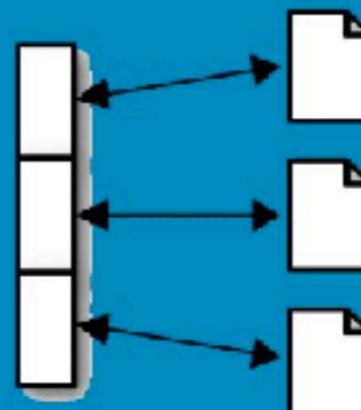


Column Family

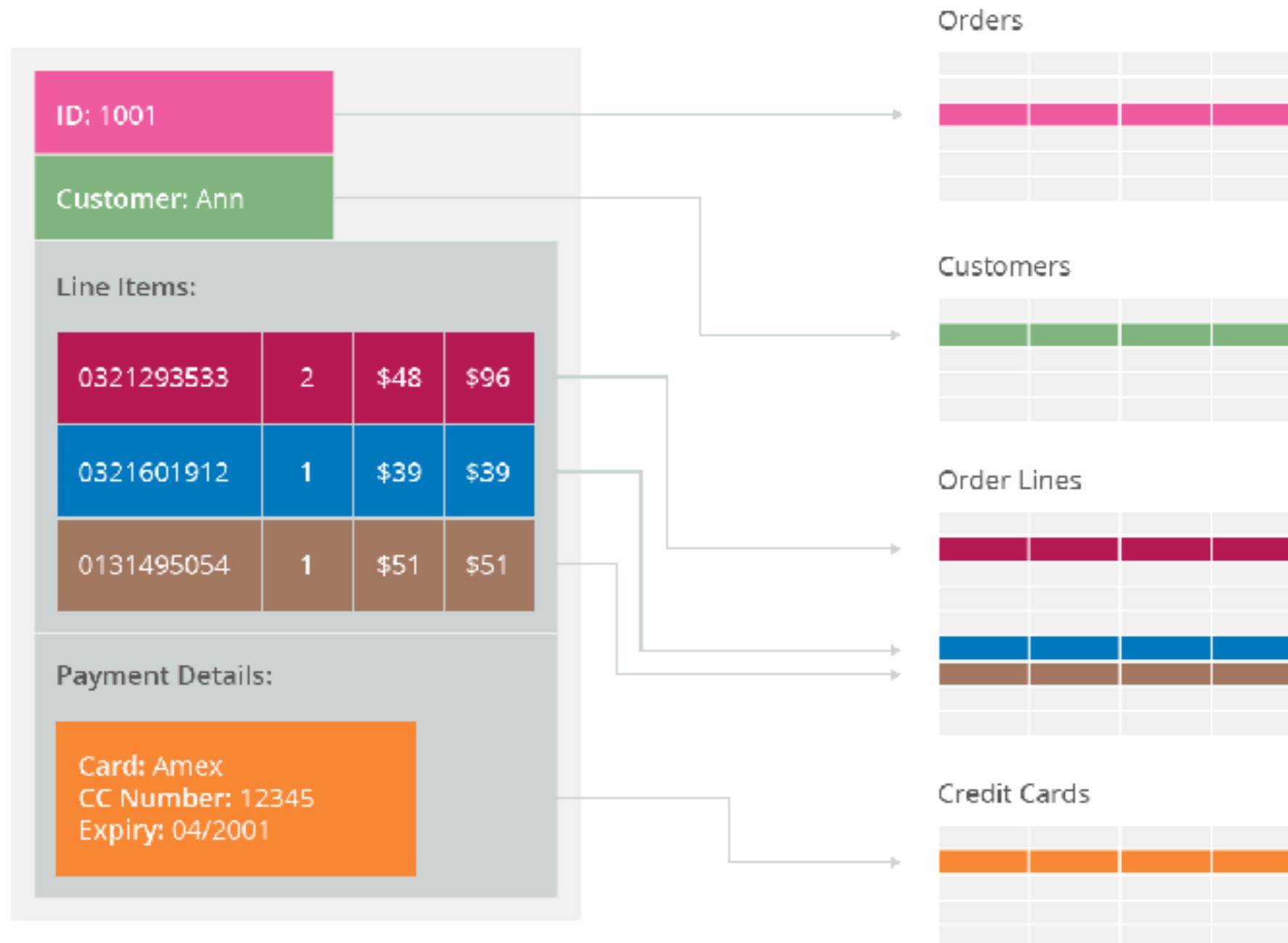


A grid structure representing a column family database. The columns are labeled with the number '1'. The first column contains values '1', '1', '1', '1', '1', '1'. The second column contains values '1', '1', '1', '1', '1', '1'. The third column contains values '1', '1', '1', '1', '1', '1'. The fourth column contains values '1', '1', '1', '1', '1', '1'. The fifth column contains values '1', '1', '1', '1', '1', '1'. The sixth column contains values '1', '1', '1', '1', '1', '1'. The seventh column contains values '1', '1', '1', '1', '1', '1'. The eighth column contains values '1', '1', '1', '1', '1', '1'. The ninth column contains values '1', '1', '1', '1', '1', '1'. The tenth column contains values '1', '1', '1', '1', '1', '1'. The bottom of the grid has a blue gradient bar.

Document



Why ?

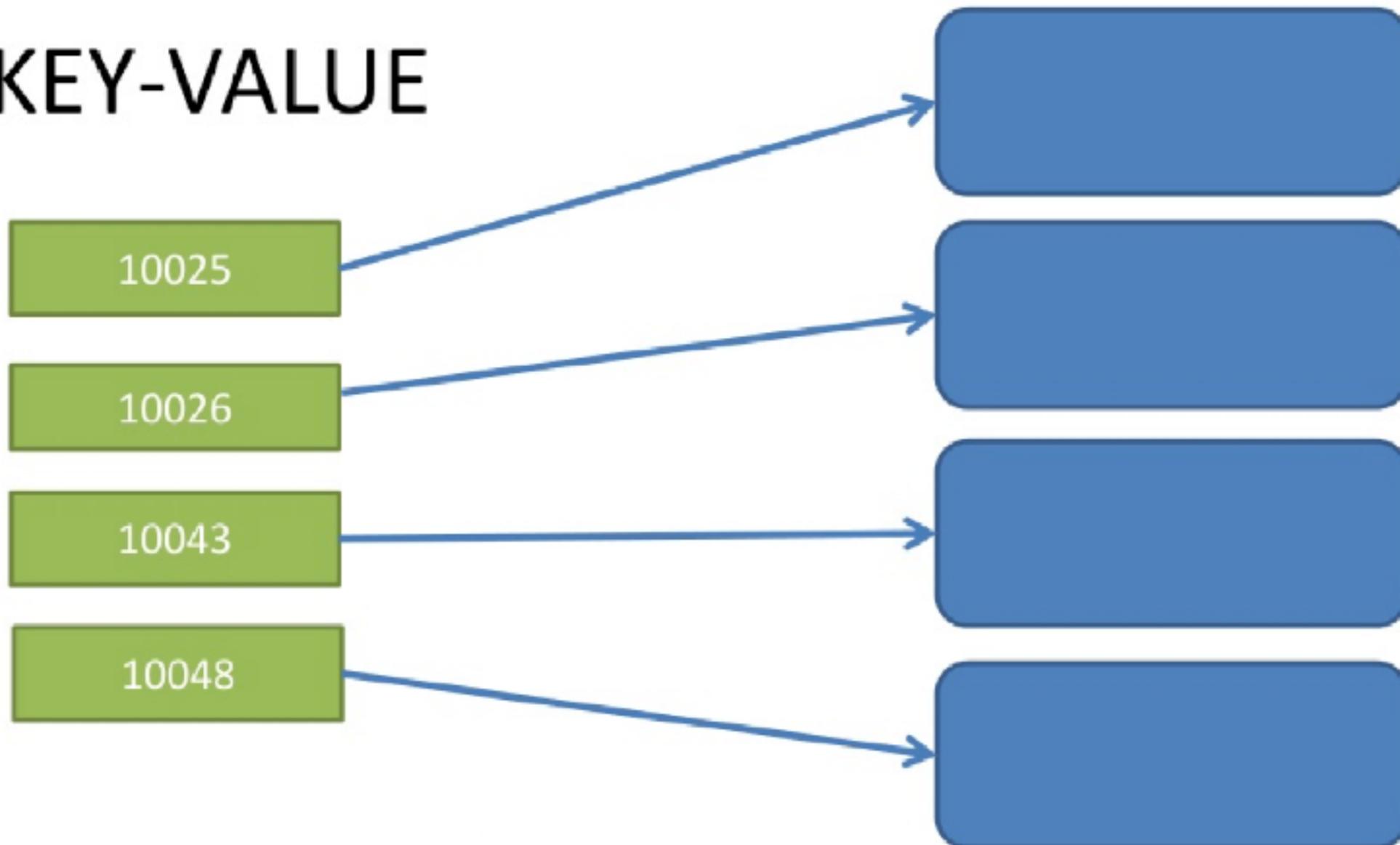


<https://www.thoughtworks.com/insights/blog/nosql-databases-overview>

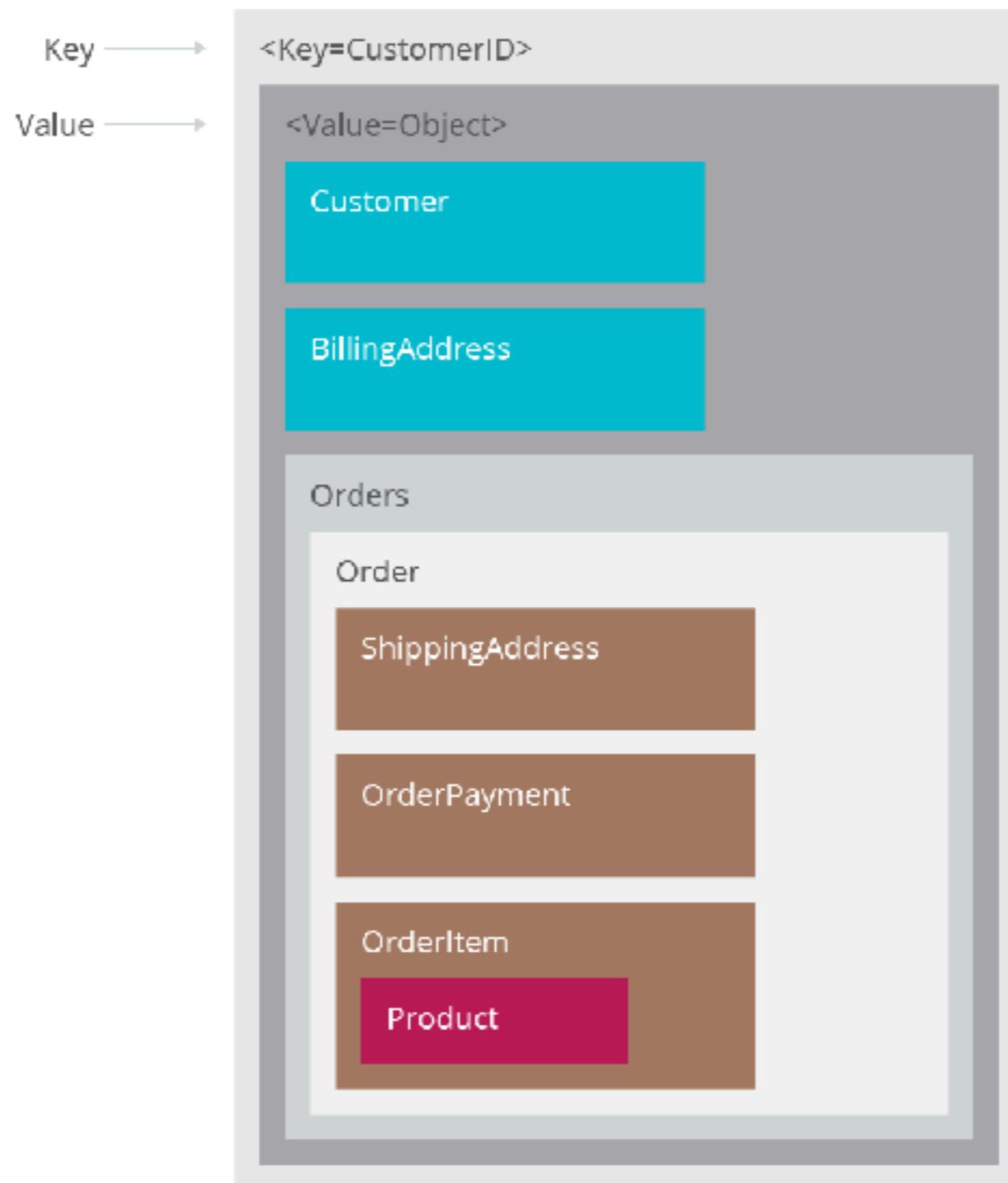


Key-value

KEY-VALUE



Key-value



Key-value use cases

Store session information
User profiles
Preferences
Shopping cart data
Caching data



Key-value tools



Amazon DynamoDB



Document

```
<Key=CustomerID>
```

```
{  
    "customerid": "fc986e48ca6" ← Key  
    "customer":  
    {  
        "firstname": "Pramod",  
        "lastname": "Sadalage",  
        "company": "ThoughtWorks",  
        "likes": [ "Biking", "Photography" ]  
    }  
    "billingaddress":  
    { "state": "AK",  
      "city": "DILLINGHAM",  
      "type": "R"  
    }  
}
```



Key-value use cases

Content management system
E-commerce system
Web analytics
Realtime analytics
Blogging platform



Document tools



mongoDB®



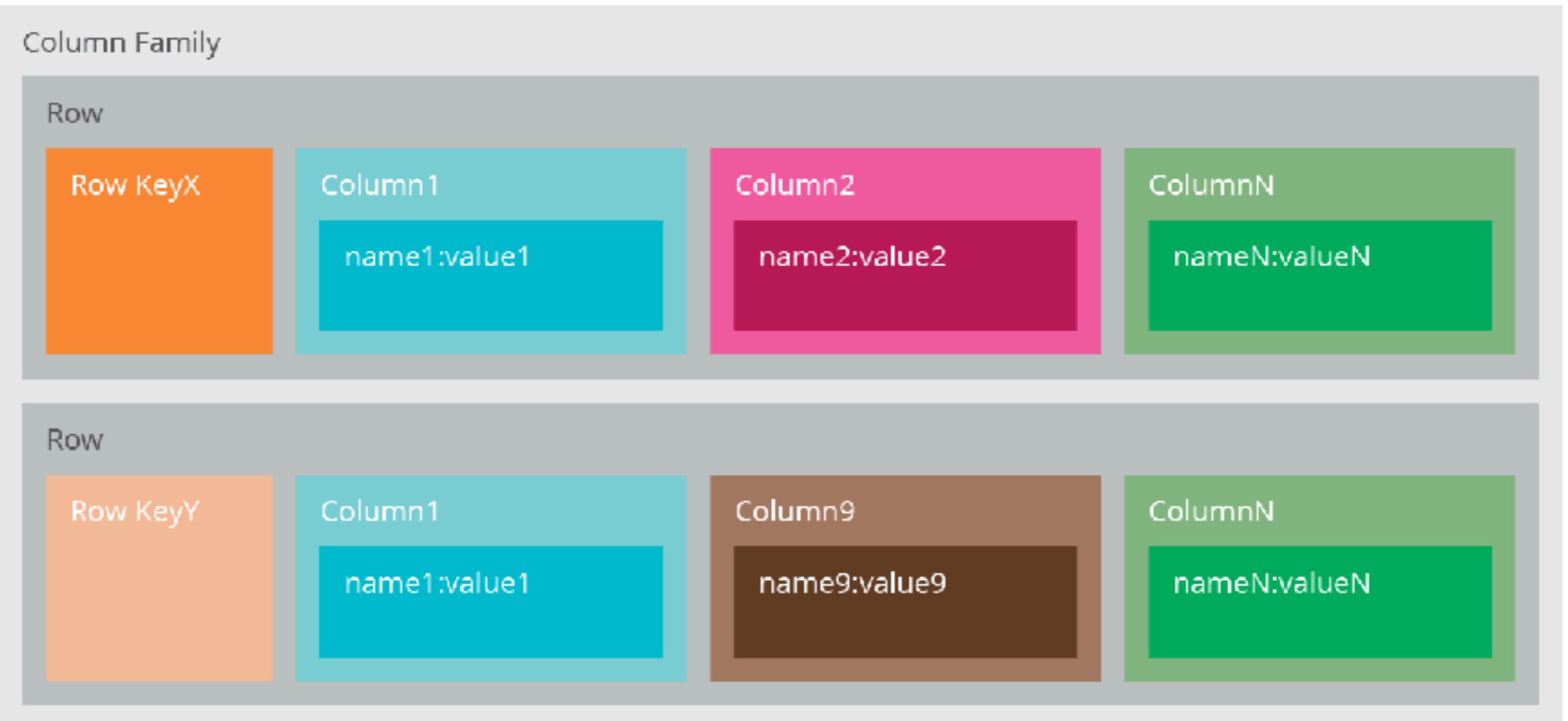
CouchDB



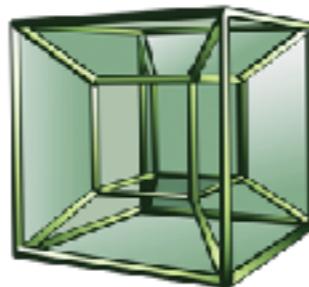
RavenDB



Column family



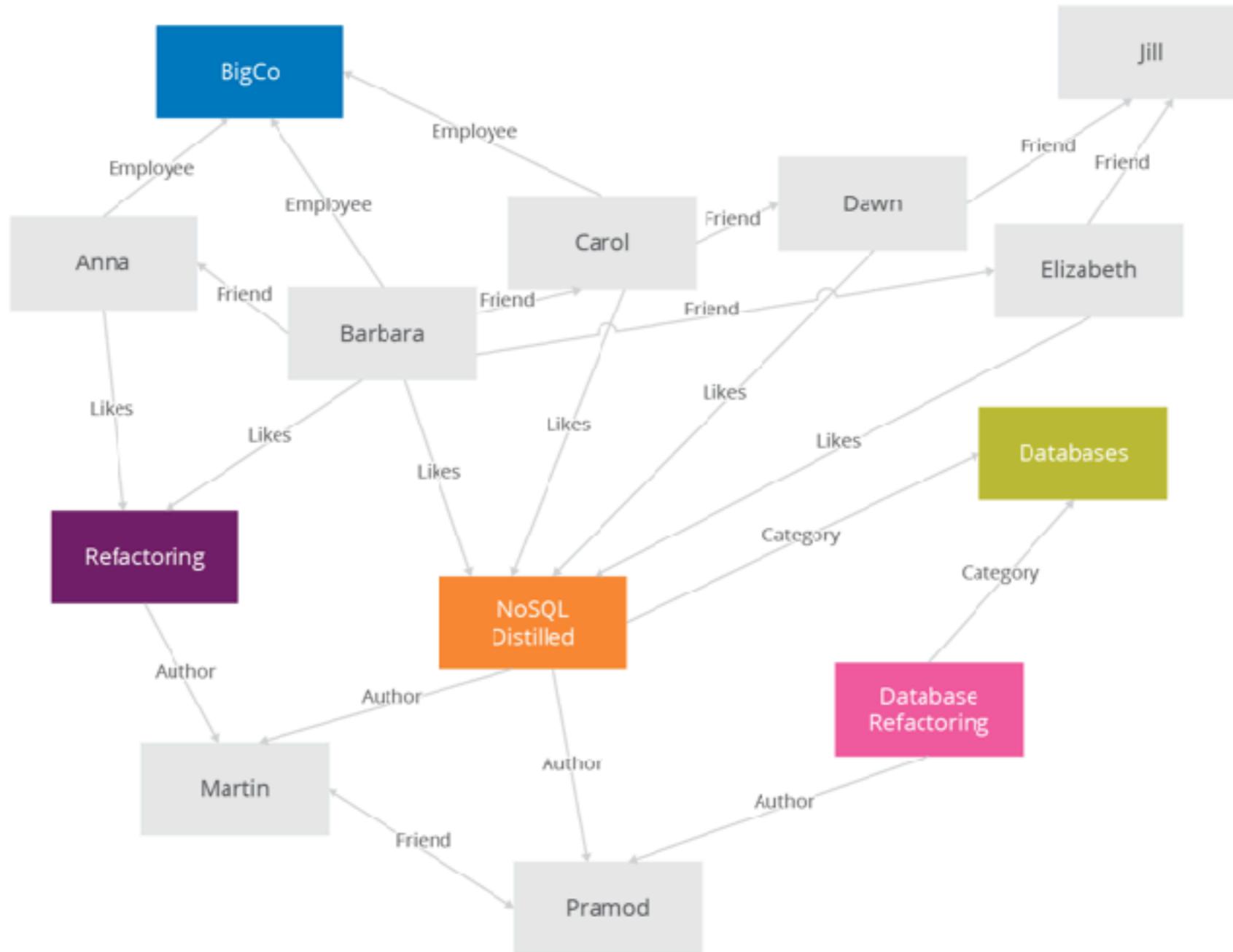
Column family tools



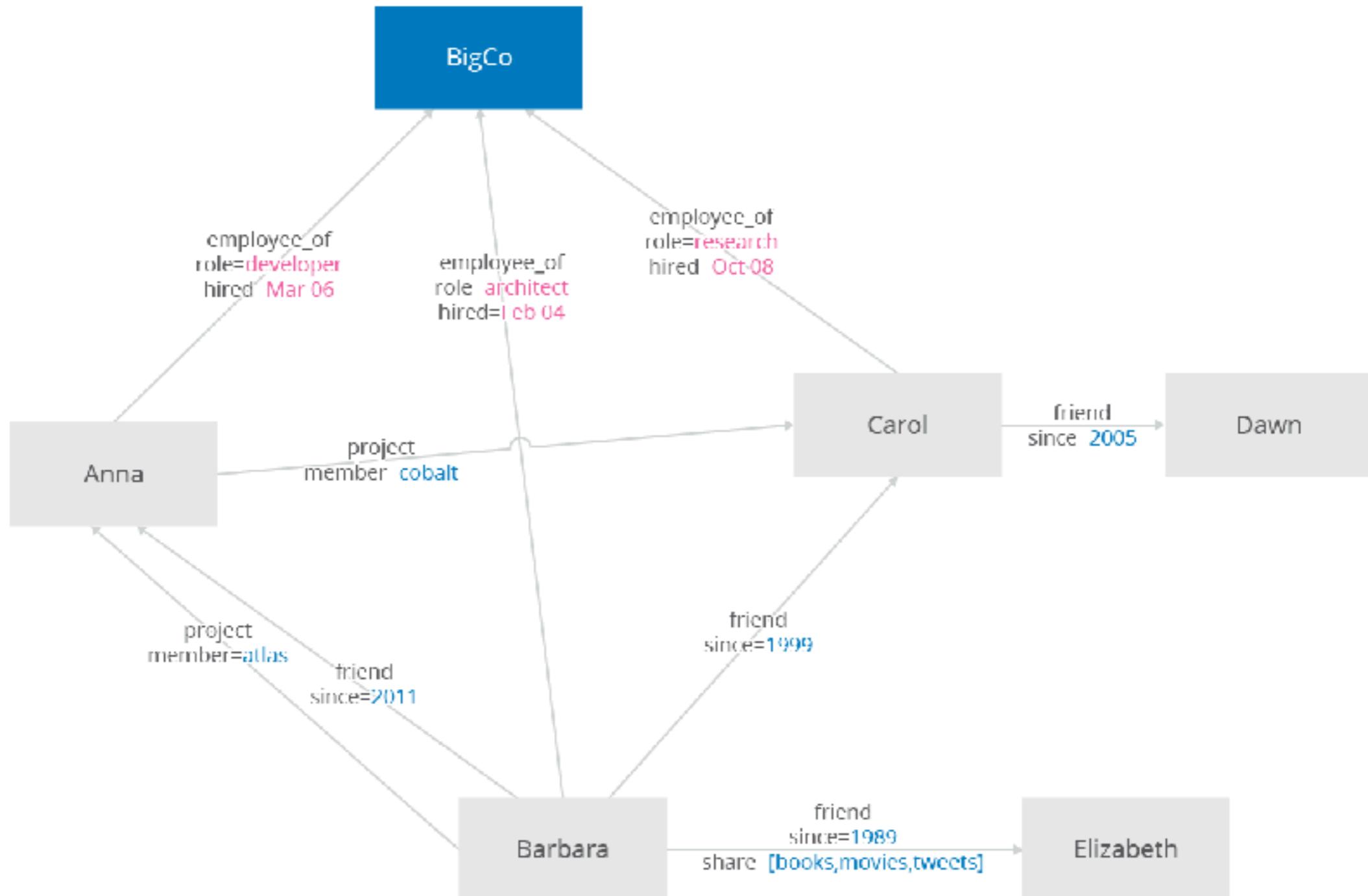
HYPERTABLE^{INC}



Graph



Graph



Graph tools



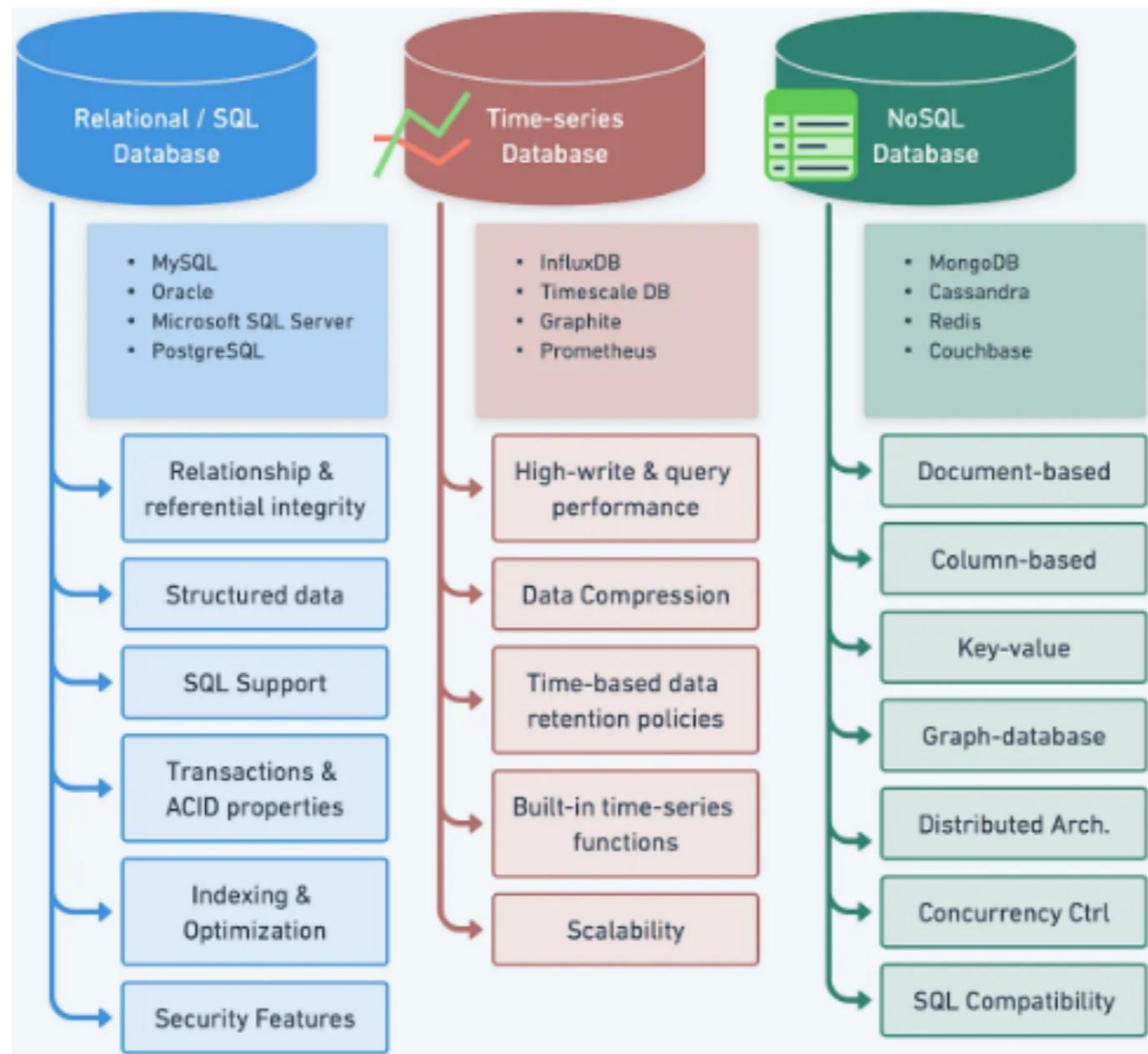
InfiniteGraph



Types of Database

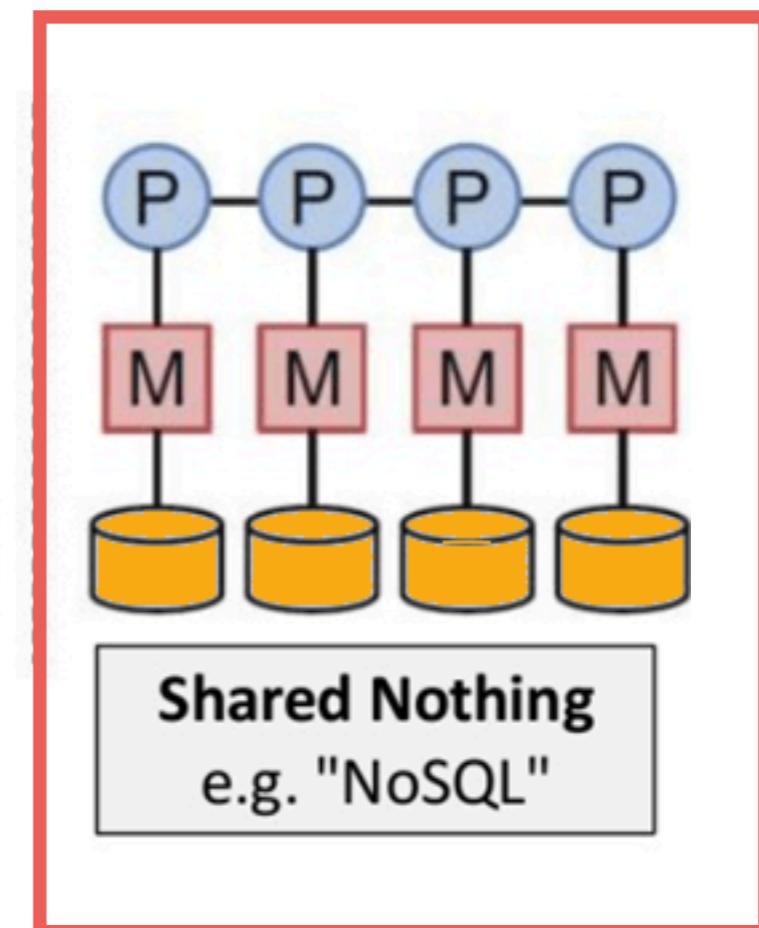
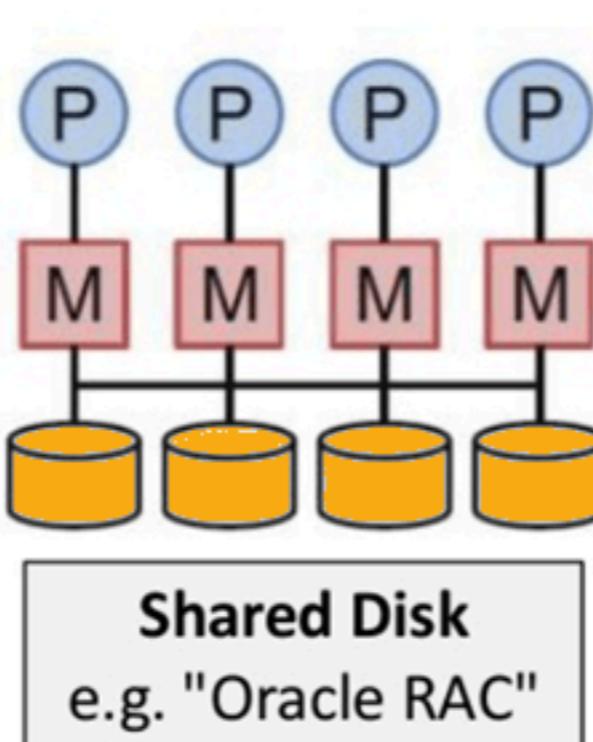
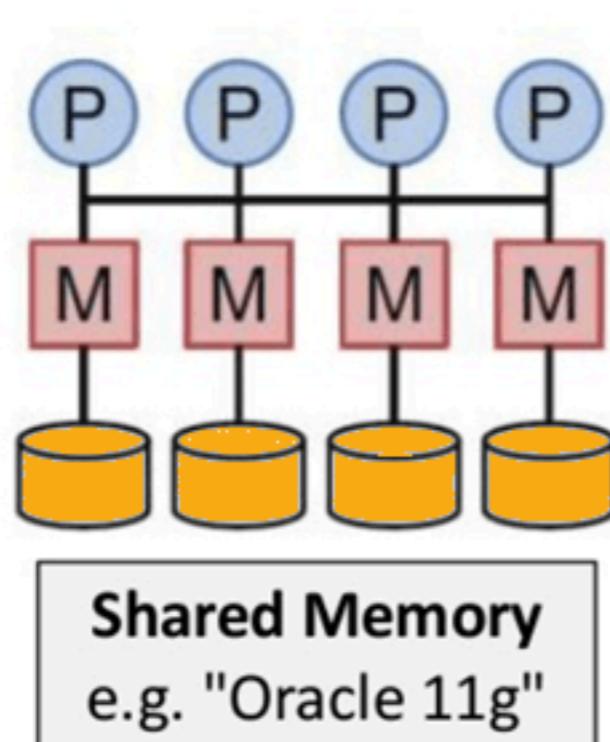


Types of Databases



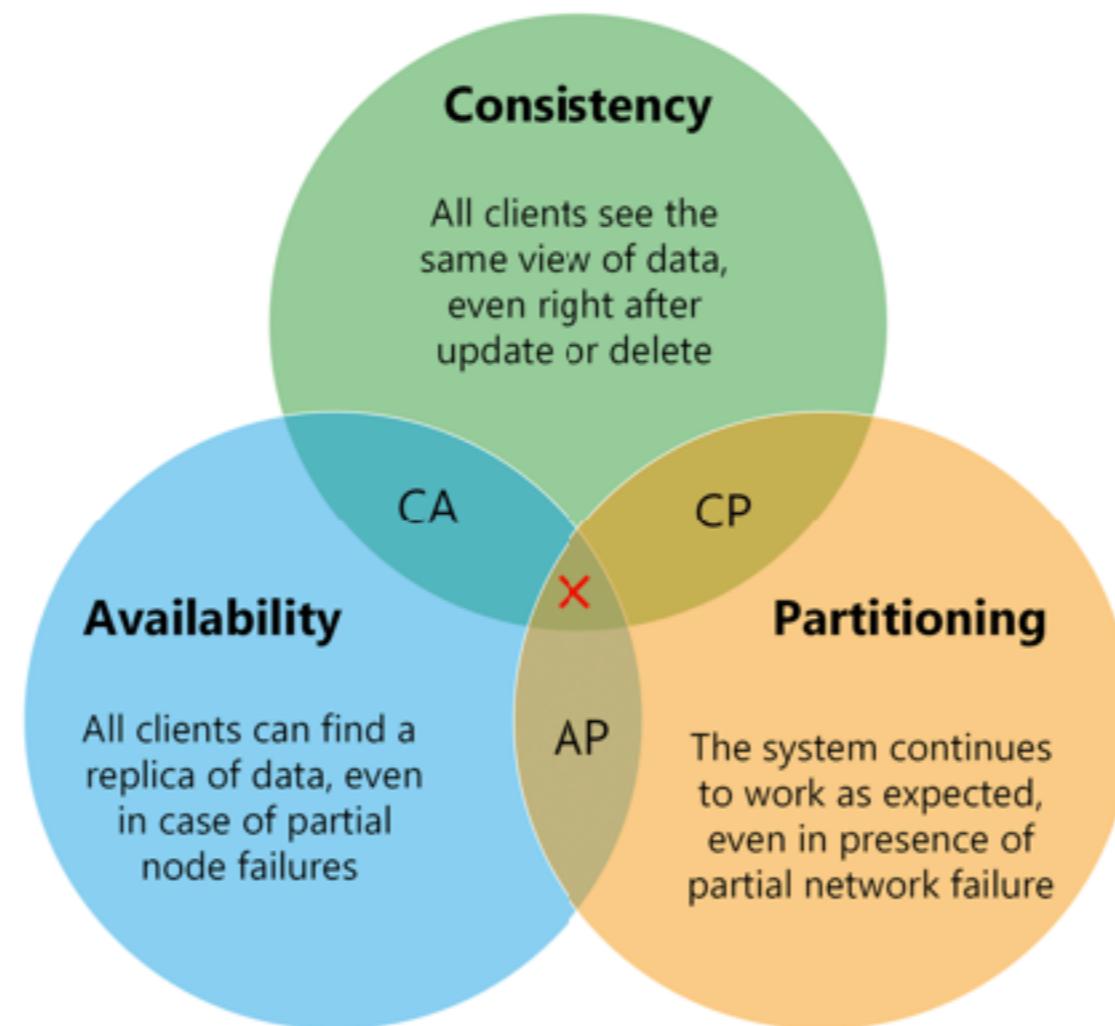
Distributed System

Shared nothing



CAP theorem

Design for distributed system



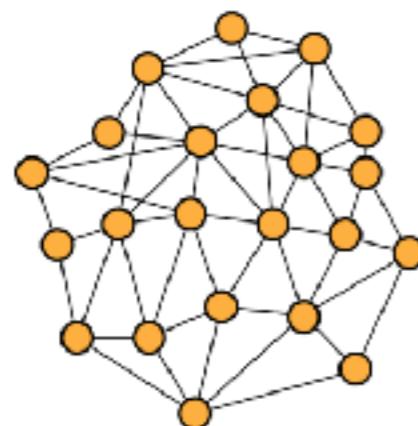
https://en.wikipedia.org/wiki/CAP_theorem



CAP theorem

C: Consistency

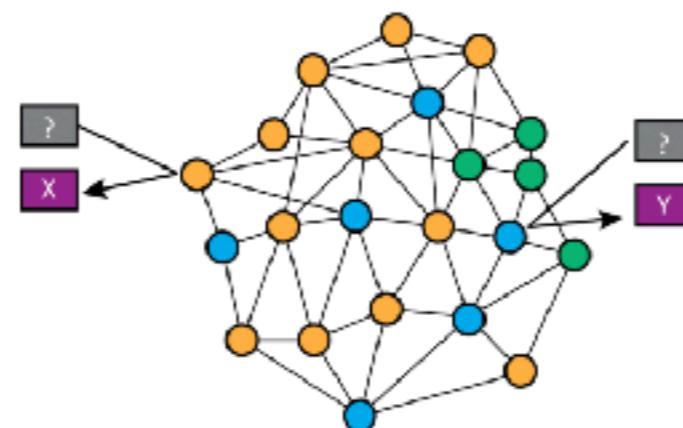
At any given time, all nodes in the network have exactly the same (most recent) **value**.



● = Value: X @ 2018-05-03T08:52:40

A : Availability

Every request to the network receives a **response**, though without any guarantee that returned data is the most recent.



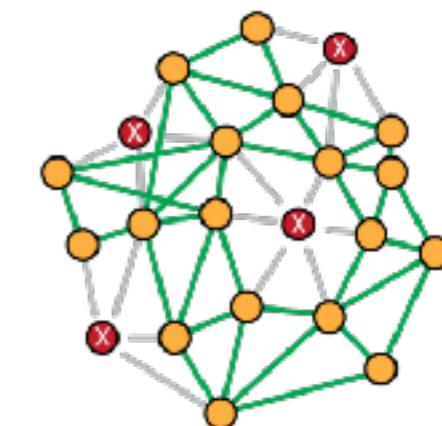
● = Value: X @ 2018-05-03T08:52:40

● = Value: Z @ 2018-05-03T08:32:58

● = Value: Y @ 2018-05-03T07:12:12

P : Partition tolerance

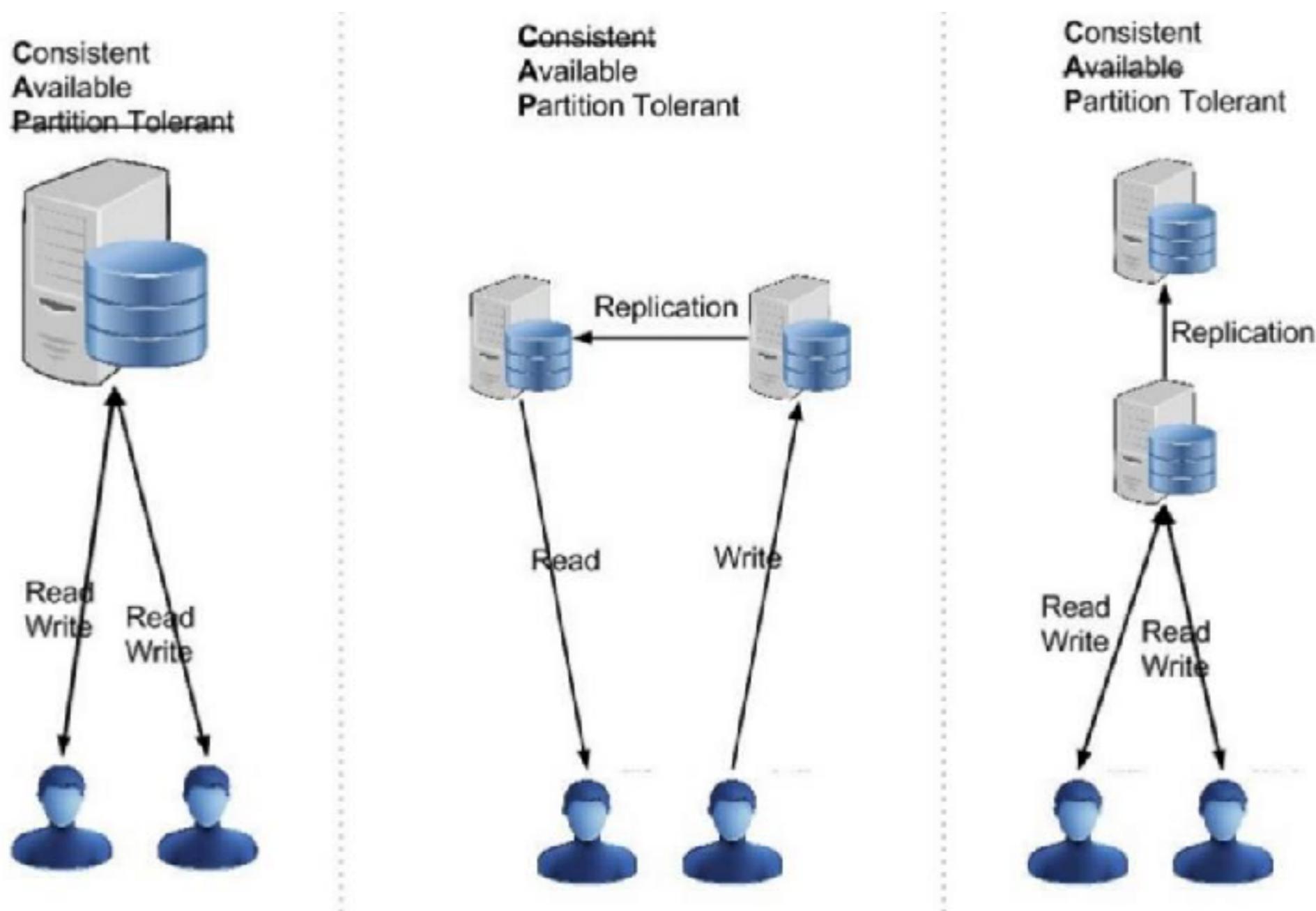
The network continues to operate, even if an arbitrary number of nodes are **failing**.



<https://cryptographics.info/cryptographics/blockchain/cap-theorem/>

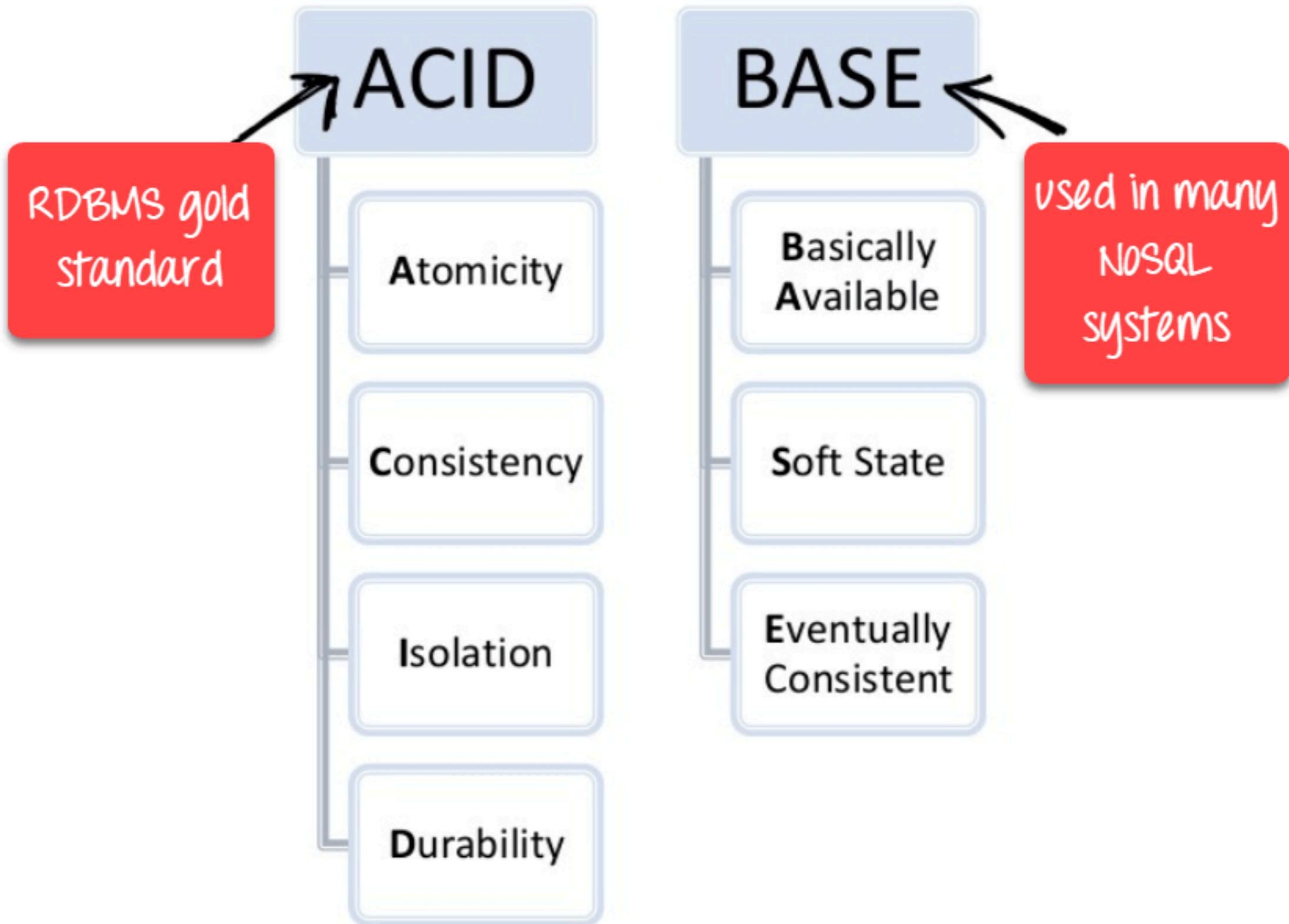


CAP theorem



ACID vs BASE







B.A.S.E

Basically Available
Soft state
Eventually consistency

“Provide flexibility approach”



Basically Available

Always work
24x7



Soft state

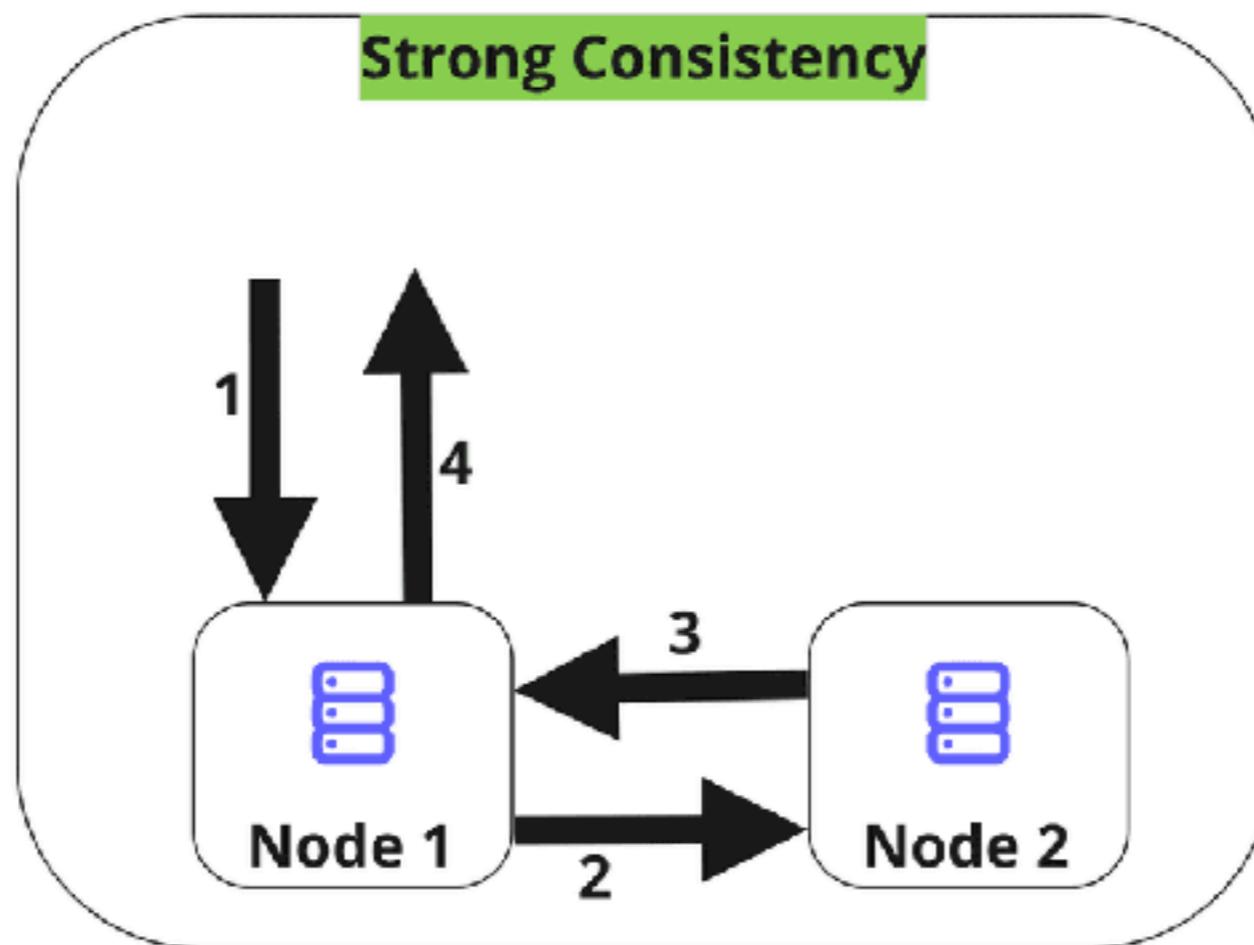
Stores don't have to be write-consistent



“Eventual Consistency”



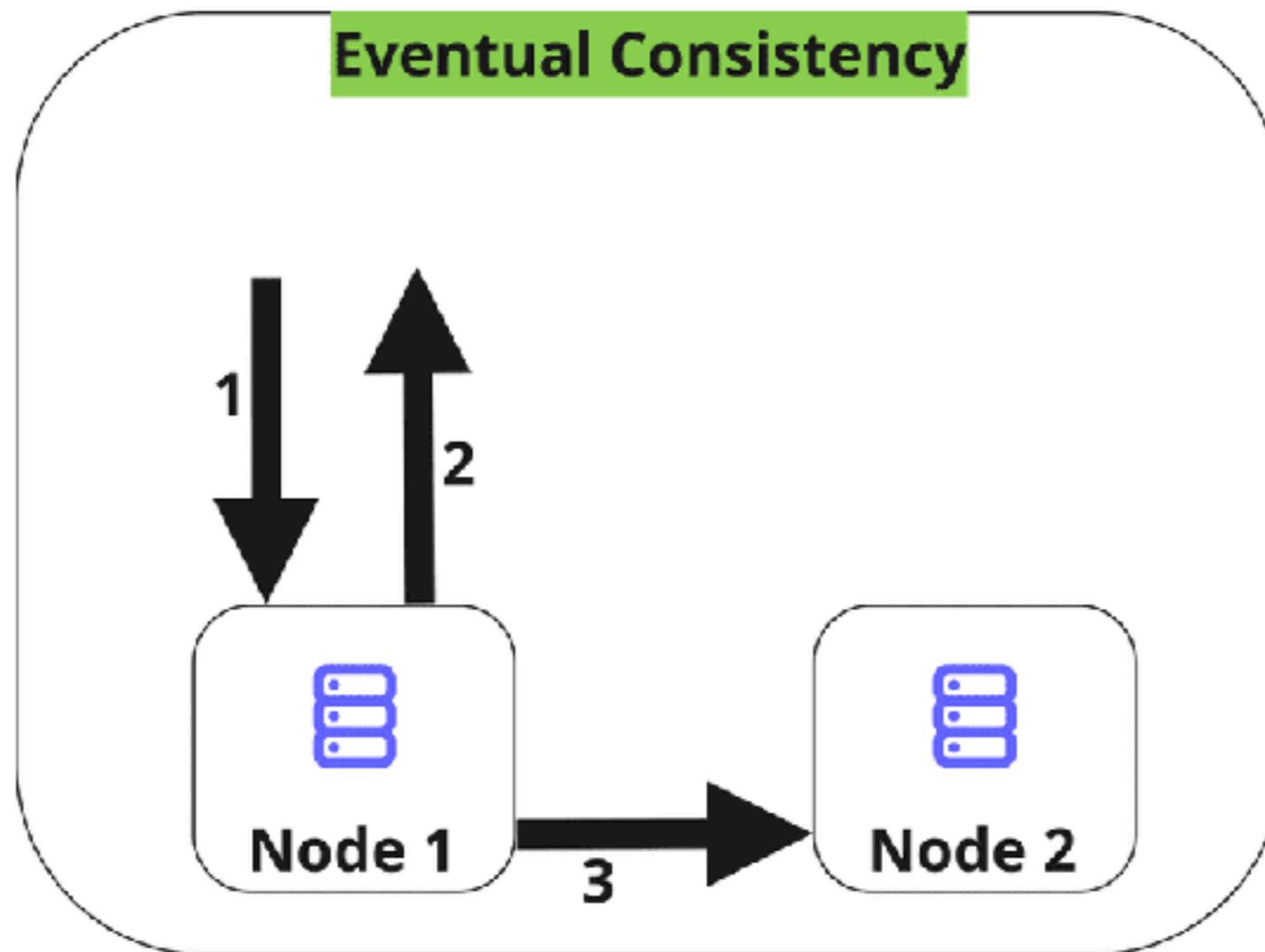
Strong Consistency



1. Write request from client.
2. Write request propagated through cluster.
3. Internal acknowledgement.
4. Acknowledgement to the client.



Eventual Consistency

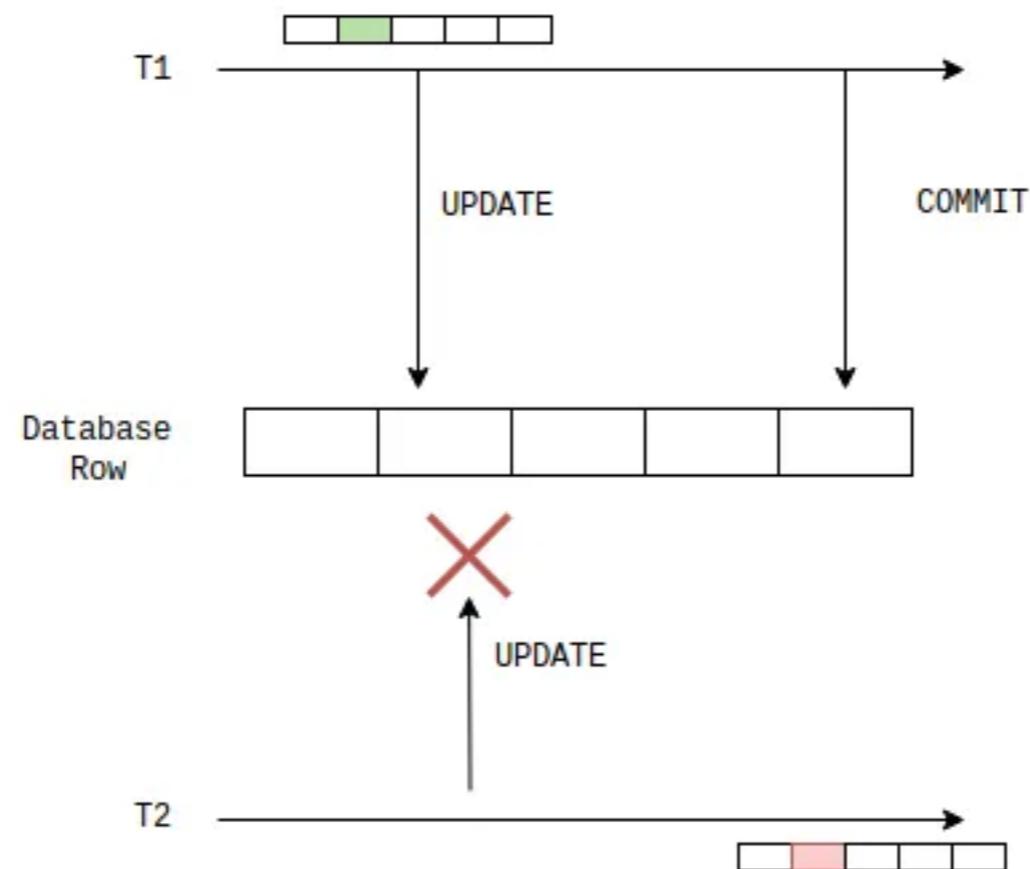


1. Write request from client.
2. Acknowledgement to the client.
3. Eventual write propagation.



Isolation level

Read uncommitted

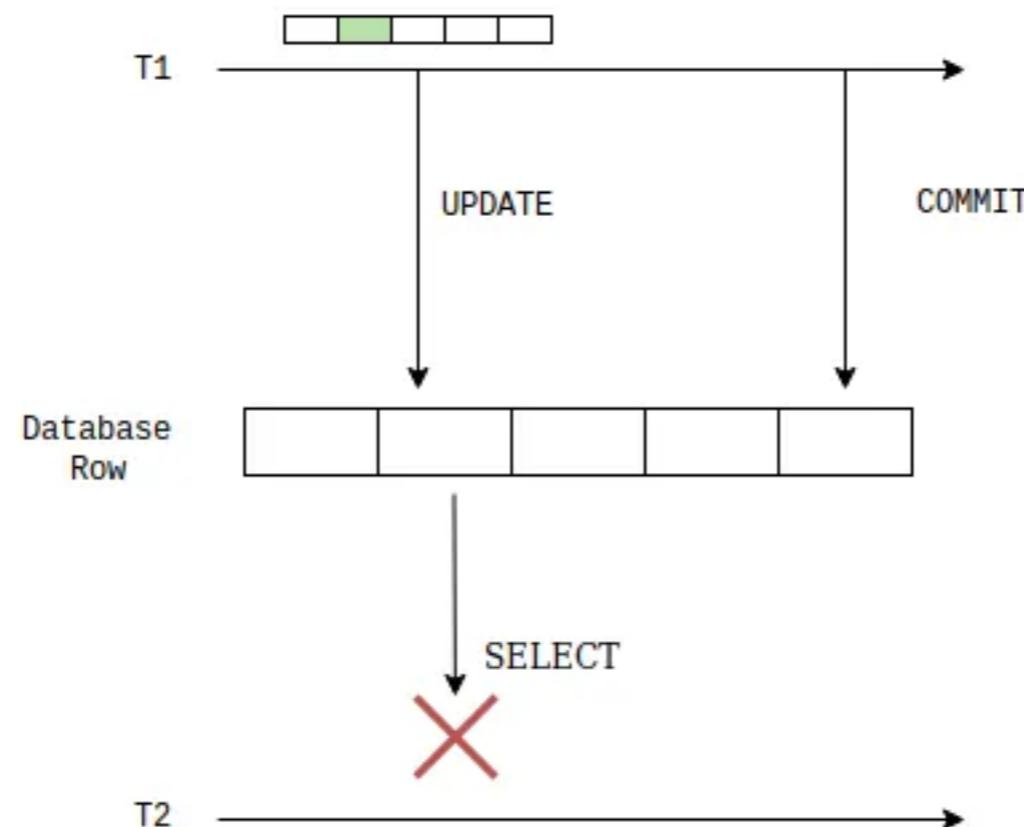


Can't update



Isolation level

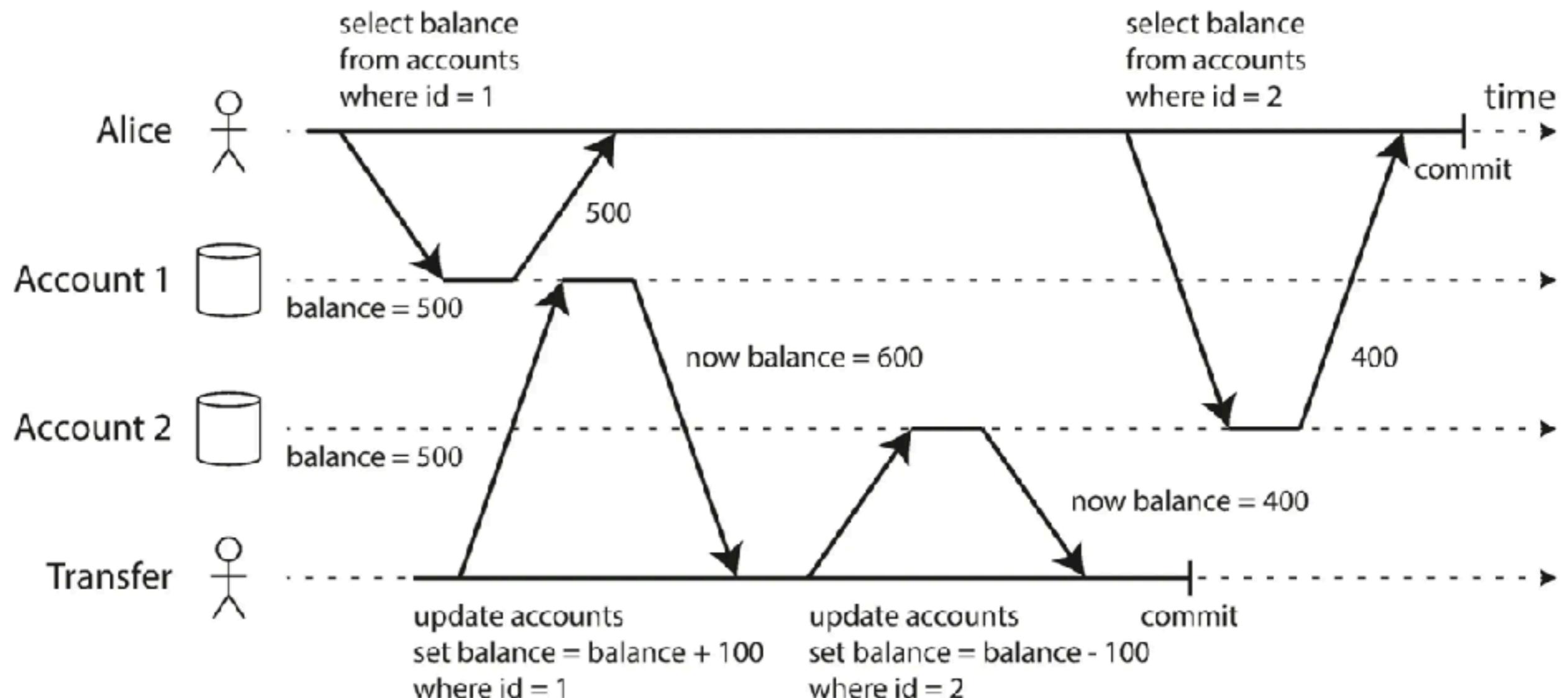
Read committed



Can't read/update



Isolation level

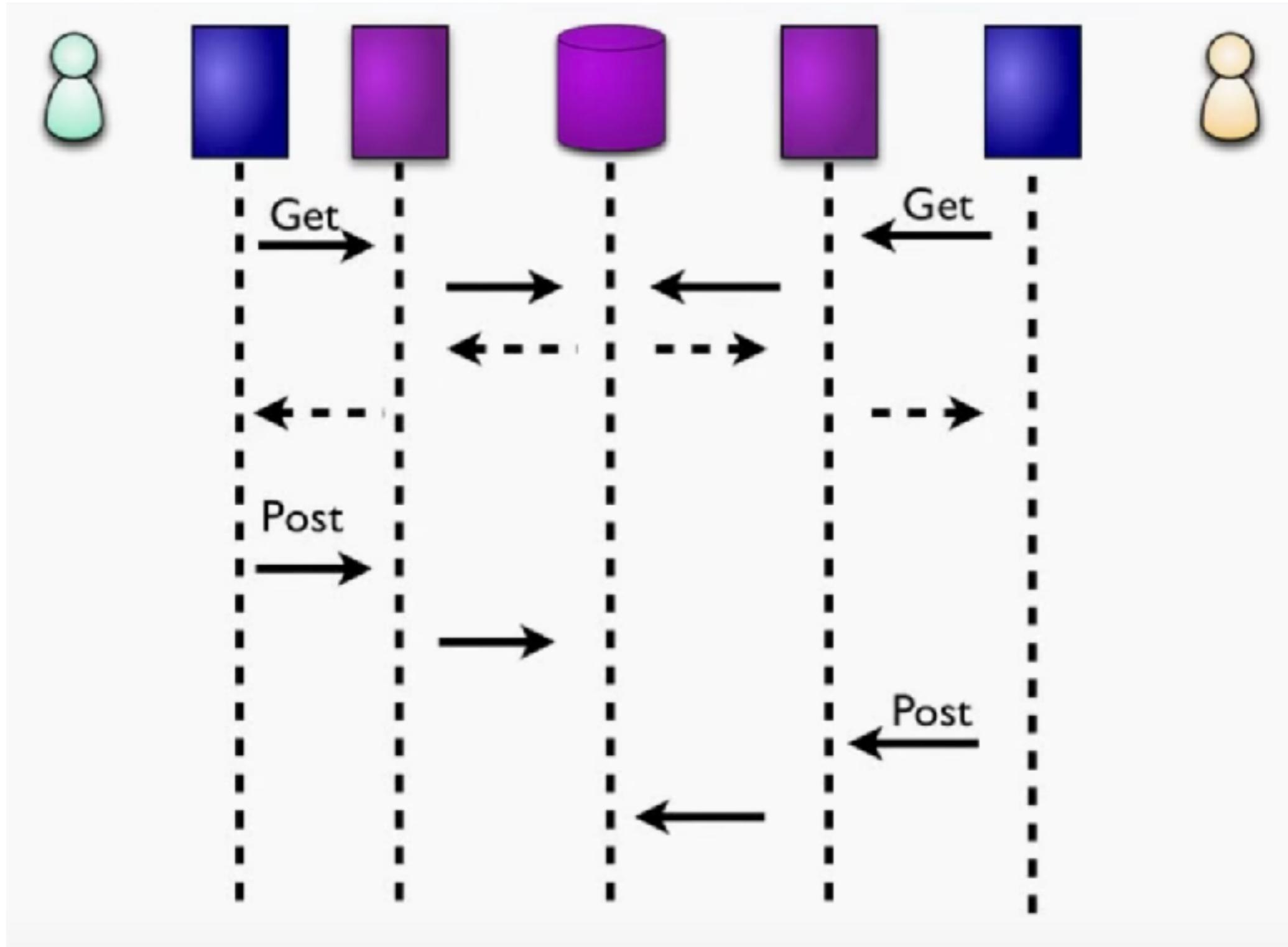


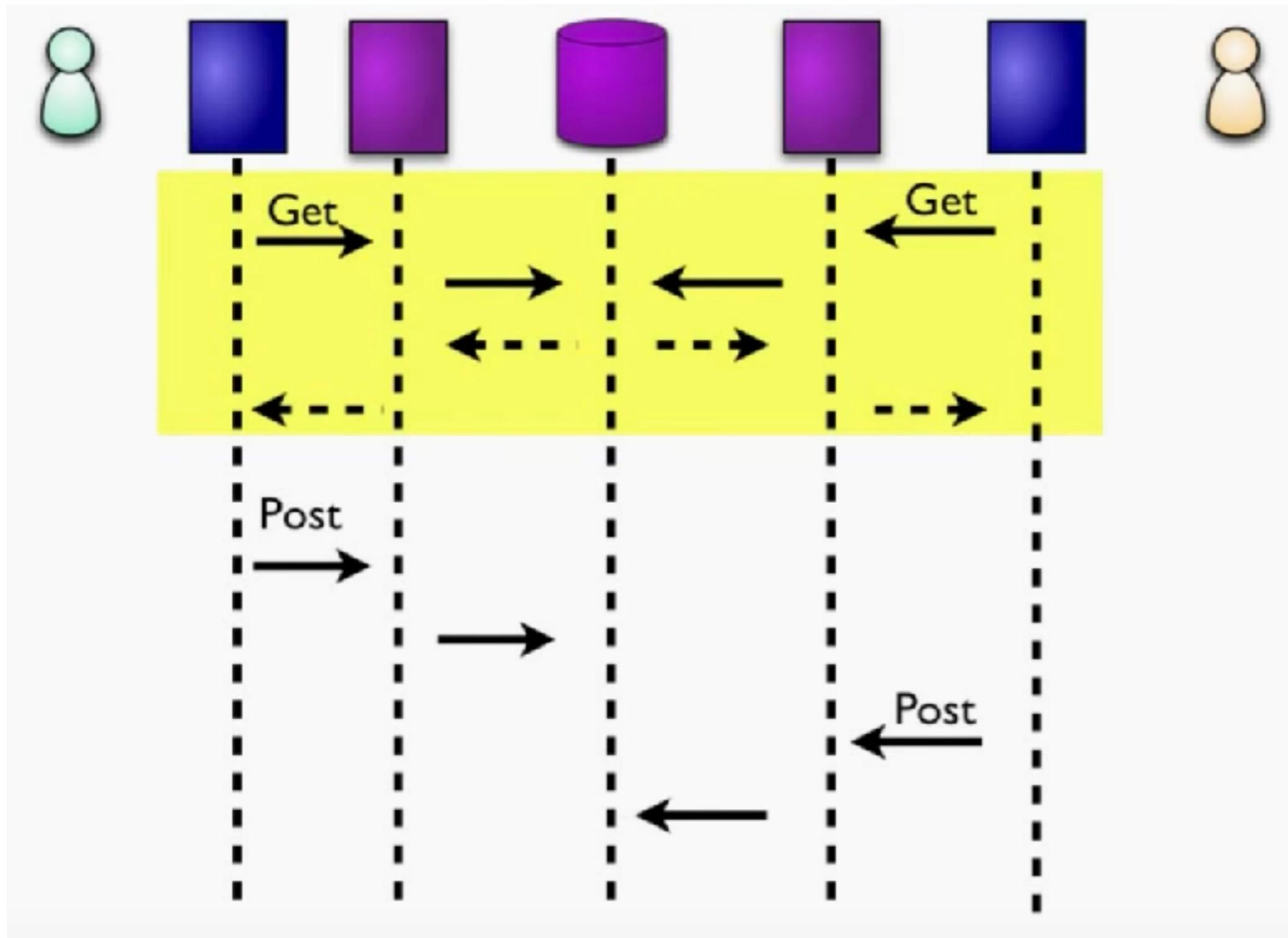
<https://medium.com/@pulkitent/system-design-database-transactions-isolation-levels-concurrency-control-contd-part-2-78db036f6971>



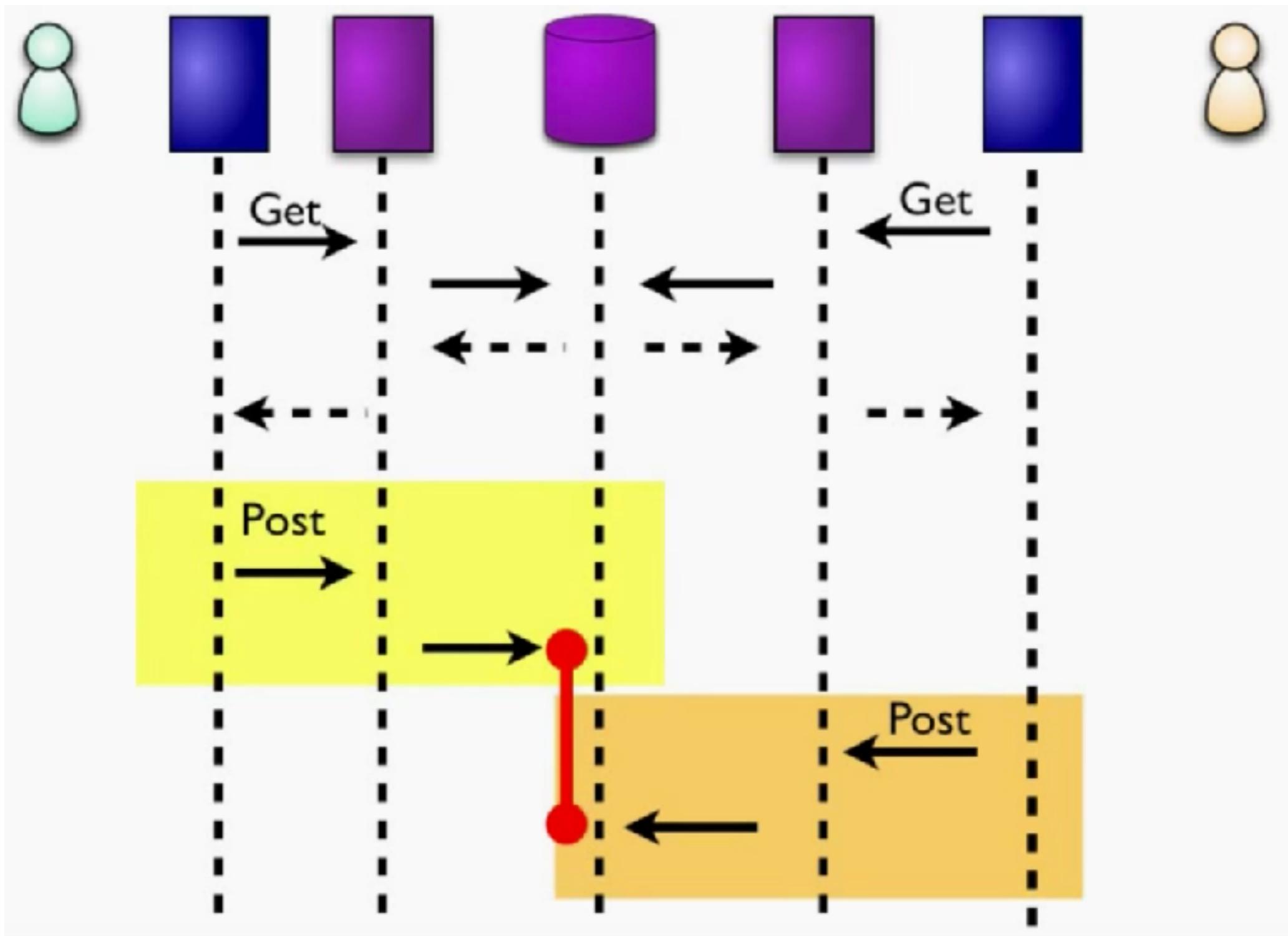
Consistency !!



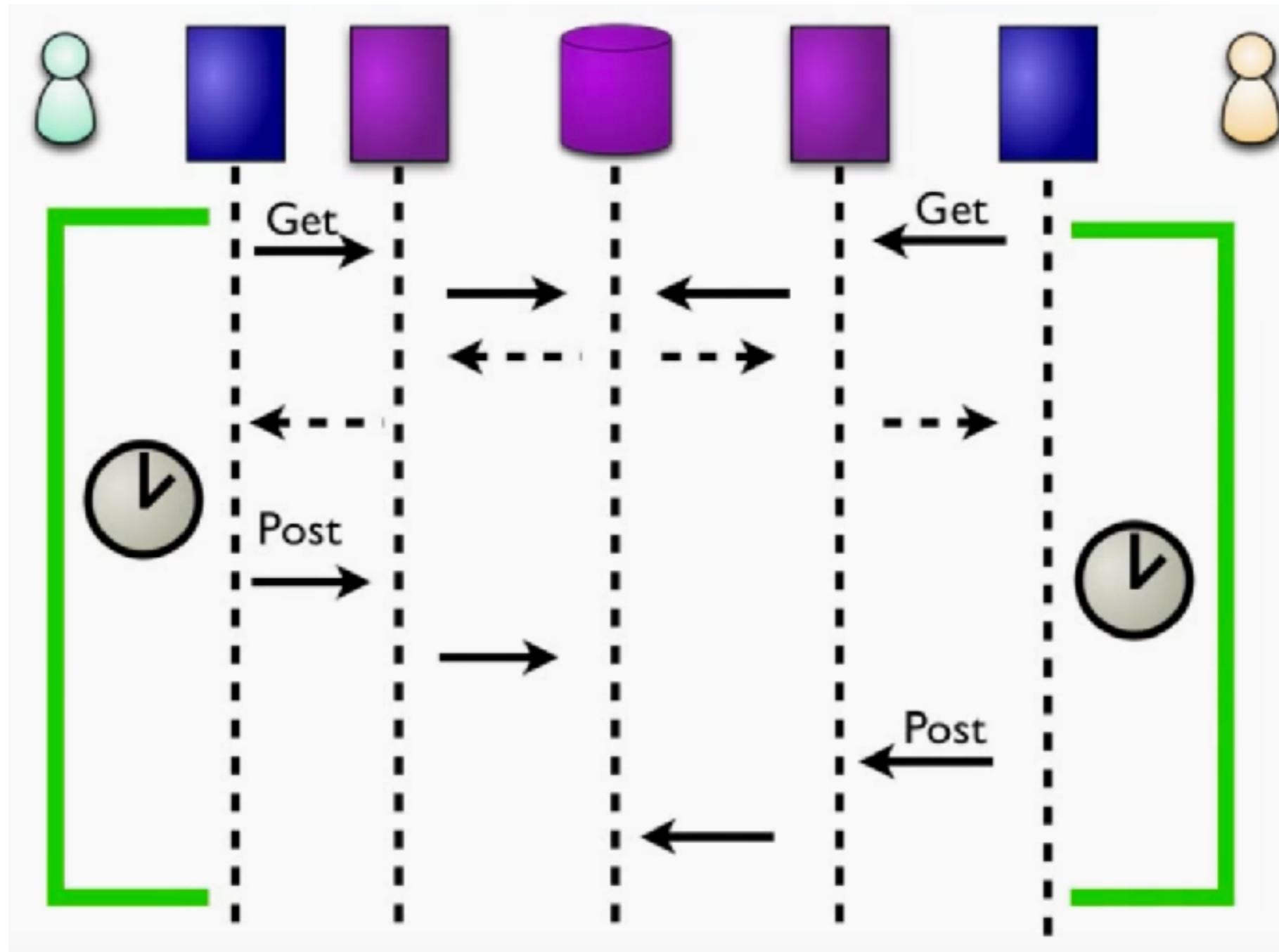




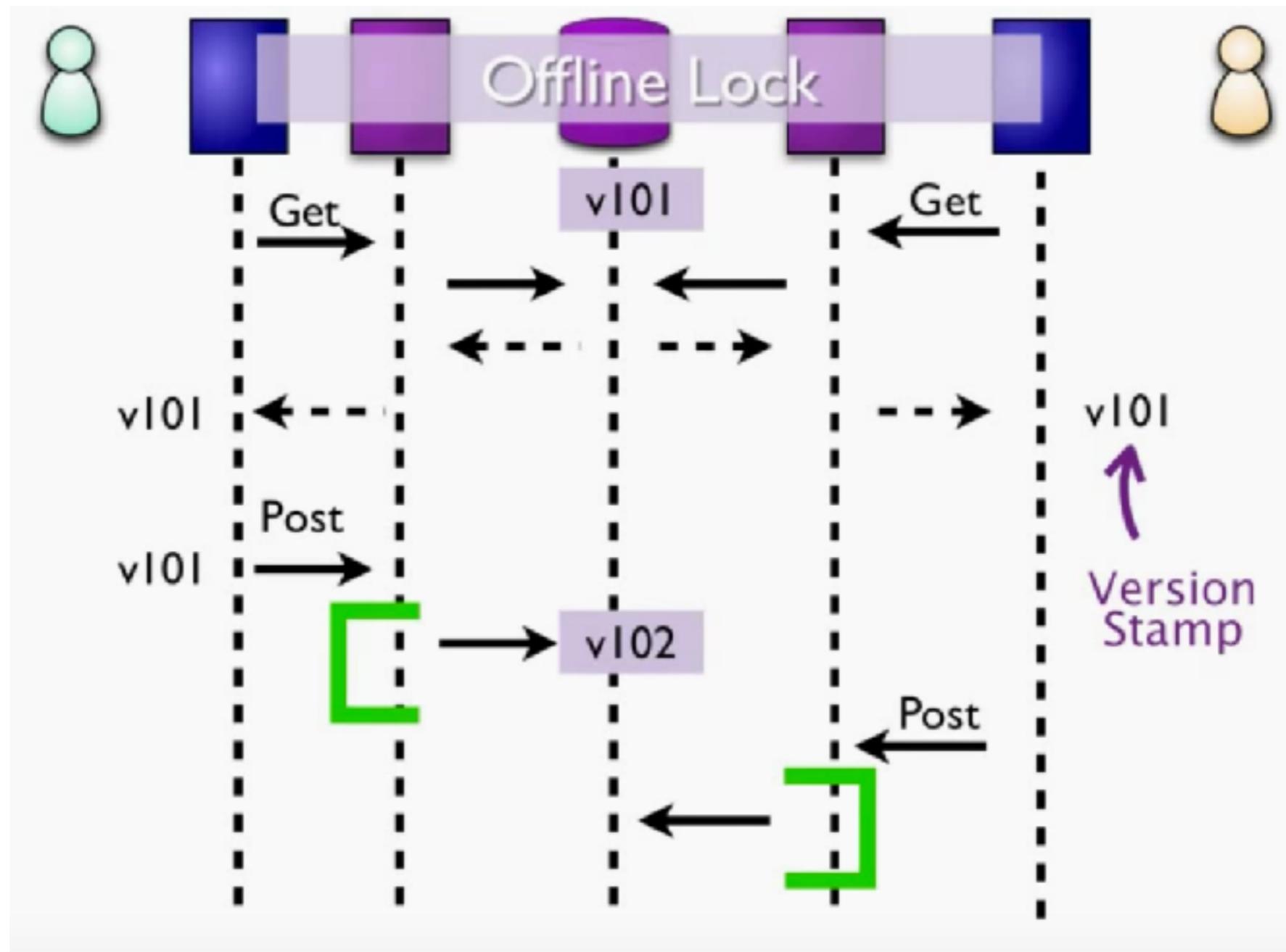
Problem ?



Transaction ?



Optimistic locking



ACID vs BASE

ACID	BASE
Provide vertical scaling	Provide horizontal scaling
Strong consistency	Weak/Eventual consistency
Isolation	Last write win, availability first
Transaction	Application managed
Available and Consistent	Available and Partition
Expensive join and relationship	Free from join and relationship

<https://systemdesignbasic.wordpress.com/2020/06/20/15-acid-vs-base-database-transactions/>



Schema-less

Dynamic schema

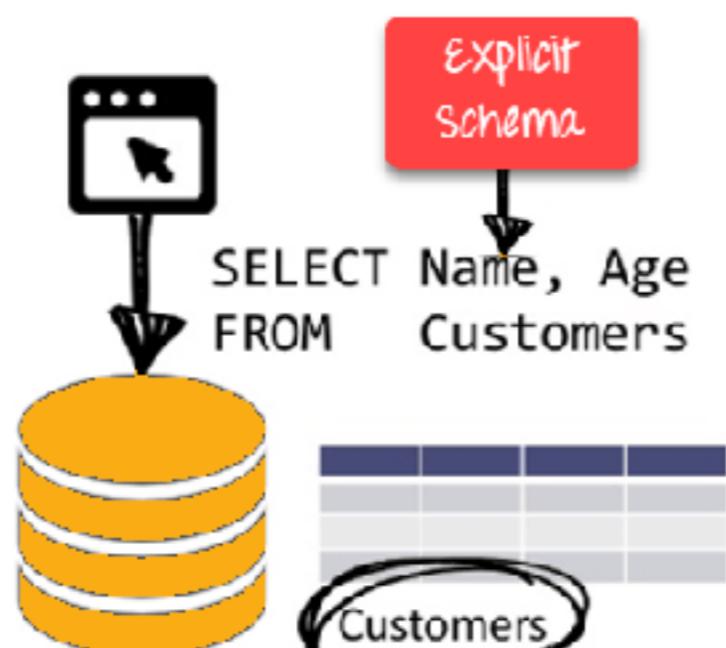
No schema



Schema-less / Dynamic schema

Offer multiple structures of data in same domain
Allow zero downtime deployment

RDBMS:

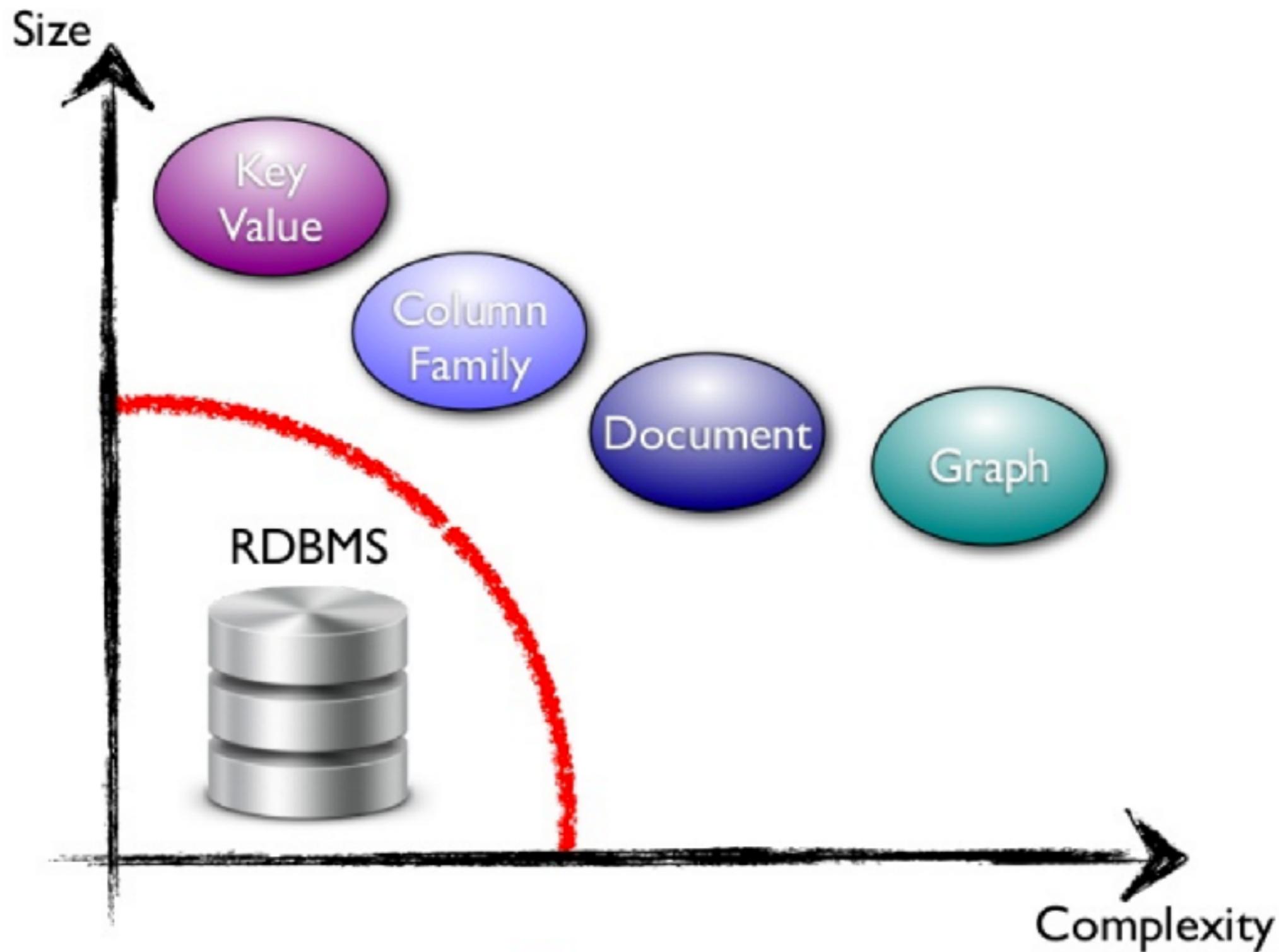


NoSQL DB:



Database selection ?





Database selection ?

Scalability
Performance
Data consistency
Data model
Security
Cost
Community and ecosystem



Design workshop



Working with MongoDB

Document database model



MongoDB Schema Design

No formal process
No algorithms
No rules



Design Consideration

Store the data

Provide good query performance

Require reasonable amount of hardware

Think about how to use data before stored !!



Database selection ?

Scalability
Performance
Data consistency
Data model
Security
Cost
Community and ecosystem



Q/A

