جامعة
القاهرة الجديدة
التكنولوجية

NEW CAIRO
TECHNOLOGICAL
UNIVERSITY

كلية تكنولوجيا الصناعة والطاقة

PEARSON BTEC International Standards Verifier

ICT Program

Dr. Amany AbdElSamea Saeed

Lec. 7 NoSQL Databases

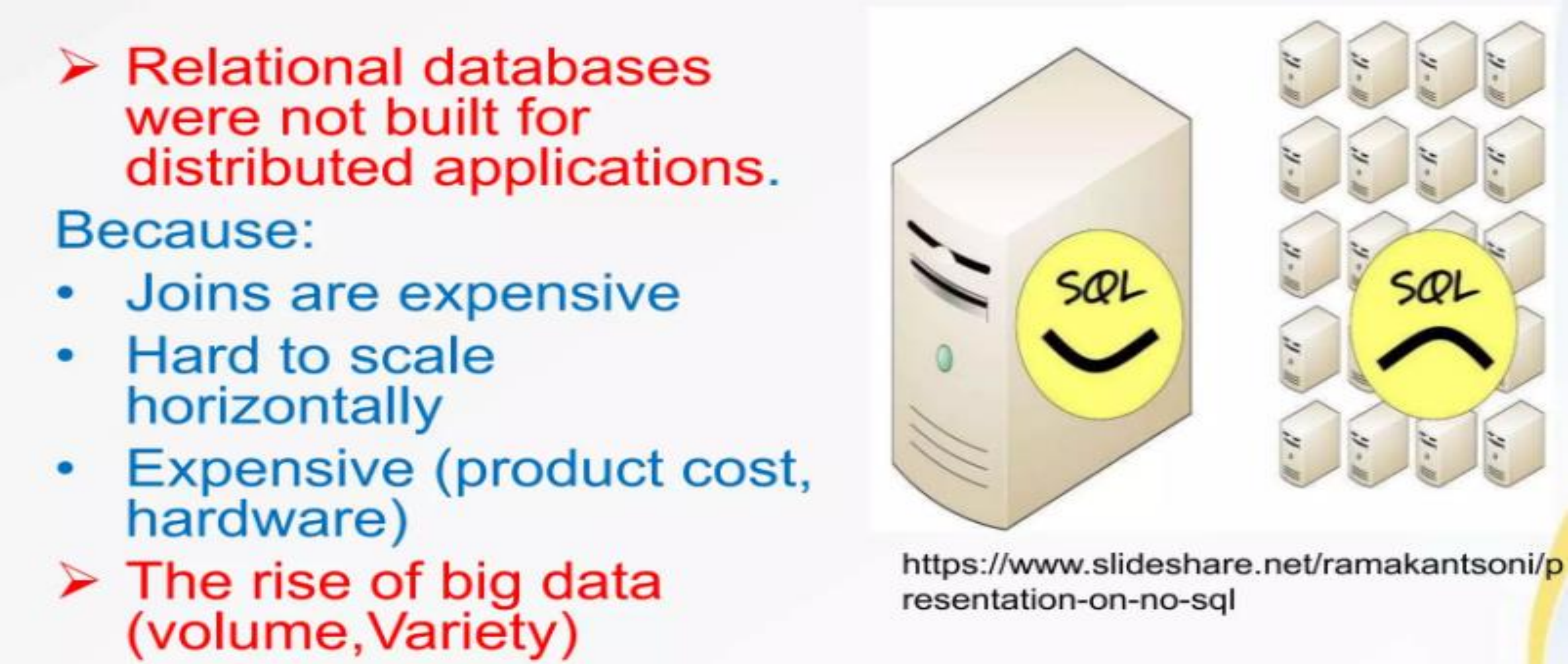
Outline

- Limitations of RDMS
- What is NoSQL?
- NoSQL characteristics
- CAP theorem
- NoSQL Types
- SQL vs. NoSQL

Relational Databases

- You can’t add a record which does not fit the schema
- You need to add NULLs to unused items in a row
- We should consider the data types. i.e. You can’t add a string to an integer field
- You can’t add multiple items in a field (You should create another table: primary-key, foreign key, joins, normalization, …..)

Limitations of RDBMS



- ➢ Relational databases were not built for distributed applications.

Because:

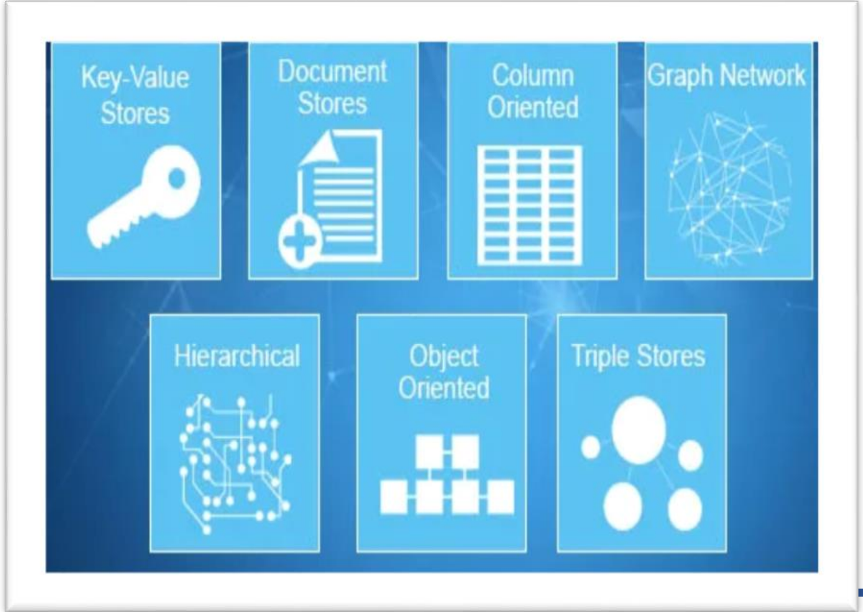
- Joins are expensive
- Hard to scale horizontally
- Expensive (product cost, hardware)

- ➢ The rise of big data (volume,Variety)

https://www.slideshare.net/ramakantsoni/presentation-on-no-sql

Need of NoSQL

- Explosion of social media sites (Facebook, Twitter, Google etc.,) with large data needs
- Rise of cloud-based solutions such as Amazon S3 (simple storage solution)
- A shift to dynamically typed data with frequent schema changes.
- Expansion of open-source community



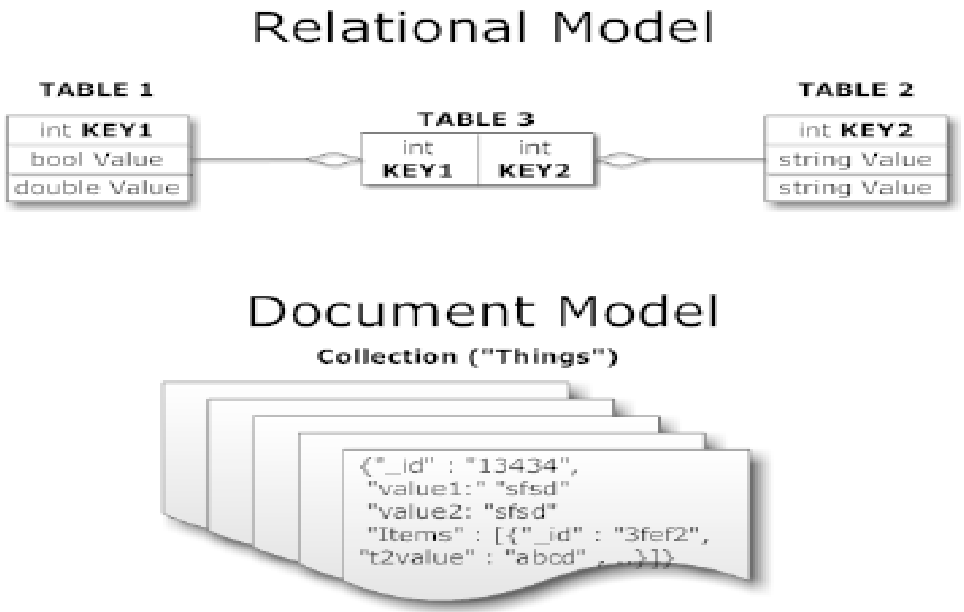
What is NOSQL?

- Stands for Not Only SQL.
- Class of non-relational data storage systems
- NoSQL database system contains various database technologies that can manage structured, unstructured, semi-structured data
- Do not require a fixed table schema nor do they use the concept of joins
- Relaxation for one or more of the ACID properties (Atomicity, Consistency, Isolation, Durability) using CAP theorem.

NOSQL Databases

In NoSQL Databases:

- There is no schema to consider
- There is no unused cell
- There is no datatype (implicit)
- Most of considerations are done in application layer
- We gather all items in an aggregate (document)



```
{"_id" : "13434",
 "value1:" "sfsd"
 "value2: "sfsd"
 "Items" : [{"_id" : "3fef2",
"t2value" : "abcd" , ...}]}
```

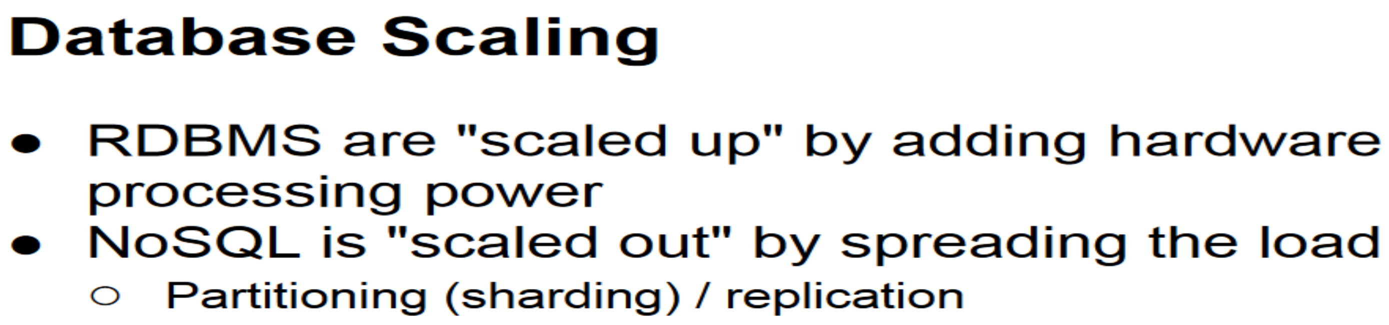
NoSQL didn't provide :

- Join
- Group By
- ACID transactions

1. SQL

NoSQL Characteristics

- Non-relational
- Flexible schema
- Other or additional query languages than SQL
- Distributed – horizontal scaling (Scaling Out)
- Less structured data
- Supports big data



Database Scaling

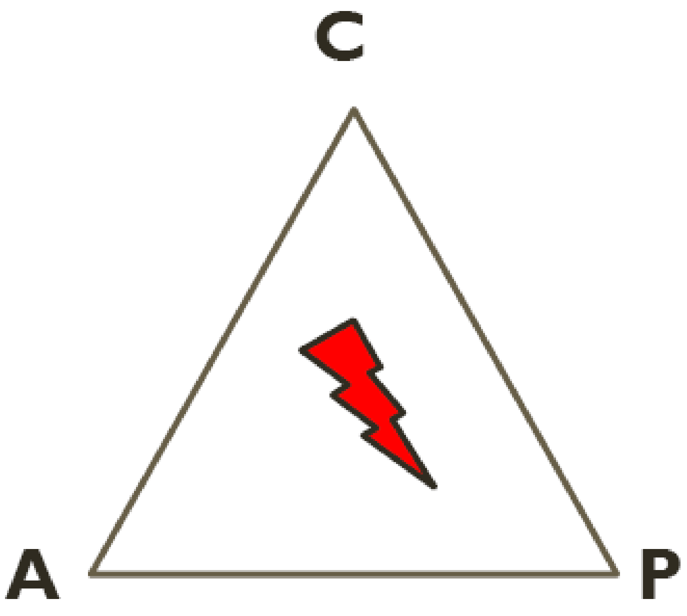
- RDBMS are "scaled up" by adding hardware processing power
- NoSQL is "scaled out" by spreading the load
  - ○ Partitioning (sharding) / replication

CAP Theorem

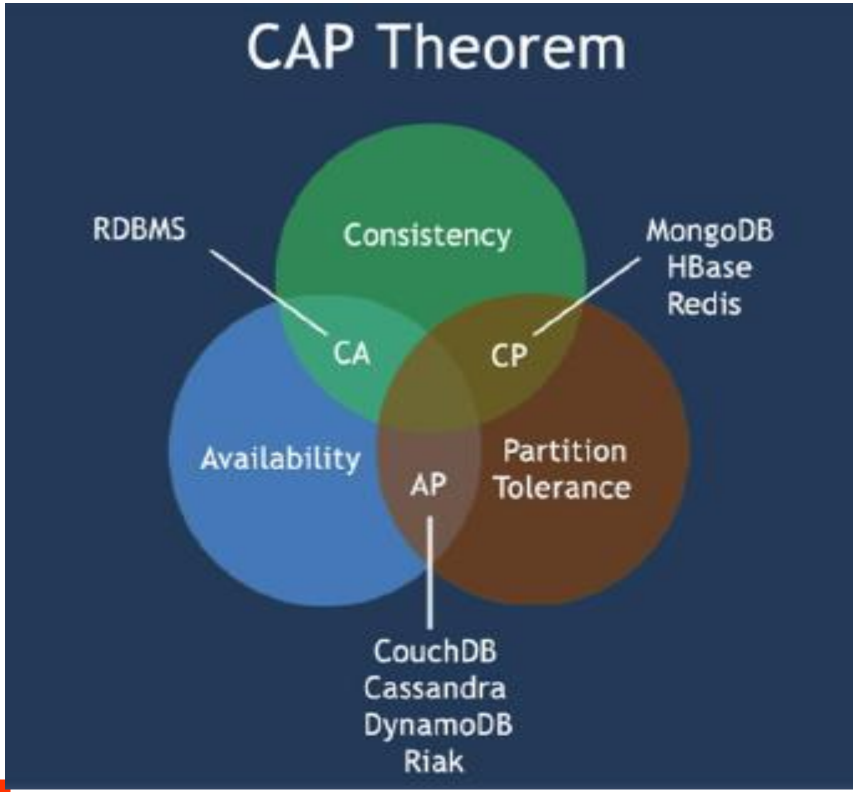
GIVEN:

- Many nodes
- Nodes contain *replicas of partitions* of the data

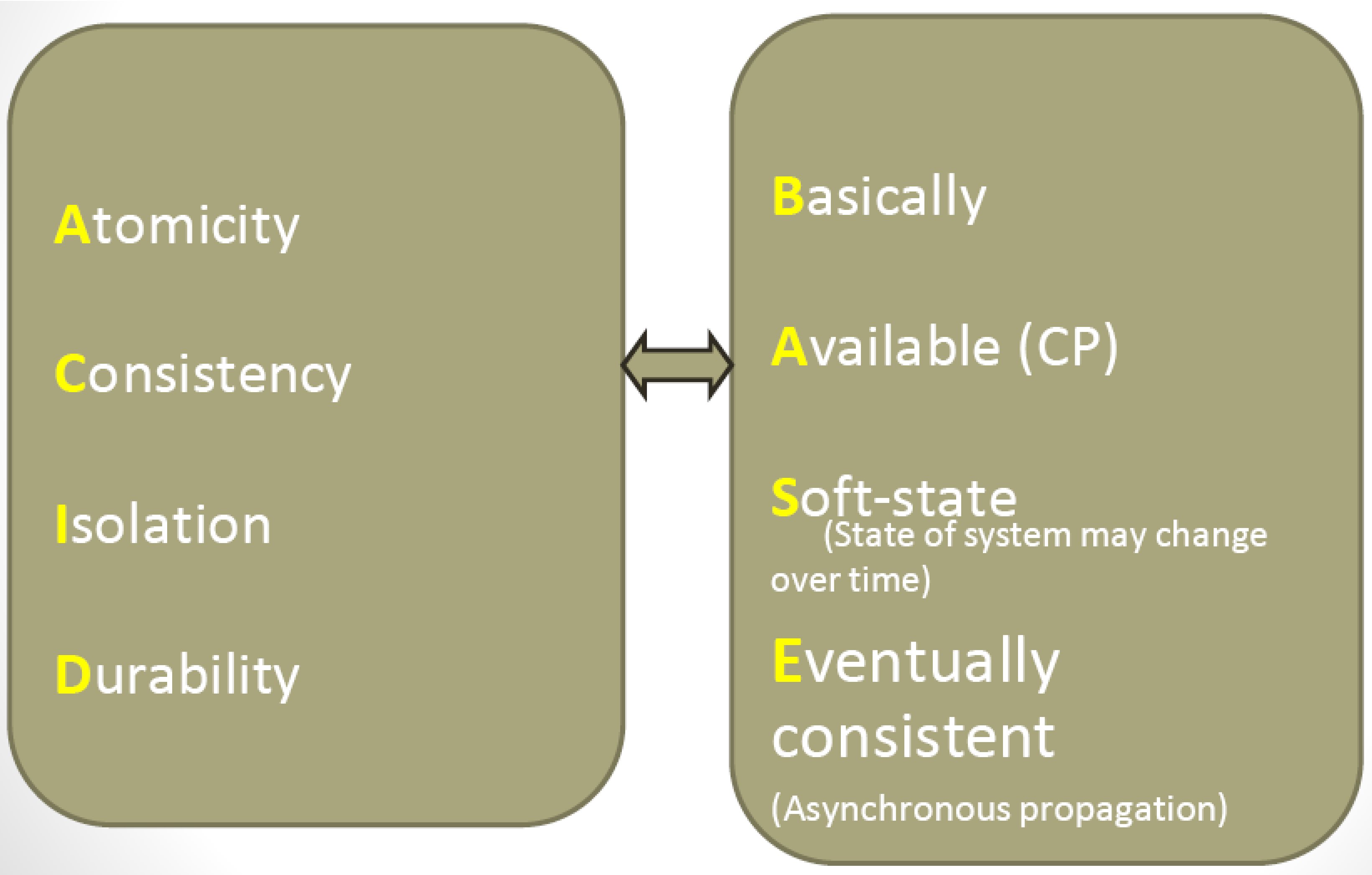
- Consistency
  - All replicas contain the same version of data
  - Client always has the same view of the data (no matter what node)
- Availability
  - System remains operational on failing nodes
  - All clients can always read and write
- Partition tolerance
  - multiple entry points
  - System remains operational on system split (communication malfunction)
  - System works well across physical network partitions



CAP Theorem: satisfying all three at the same time is impossible



RDB ACID to NoSQL BASE



Atomicity

Consistency

Isolation

Durability

Basically

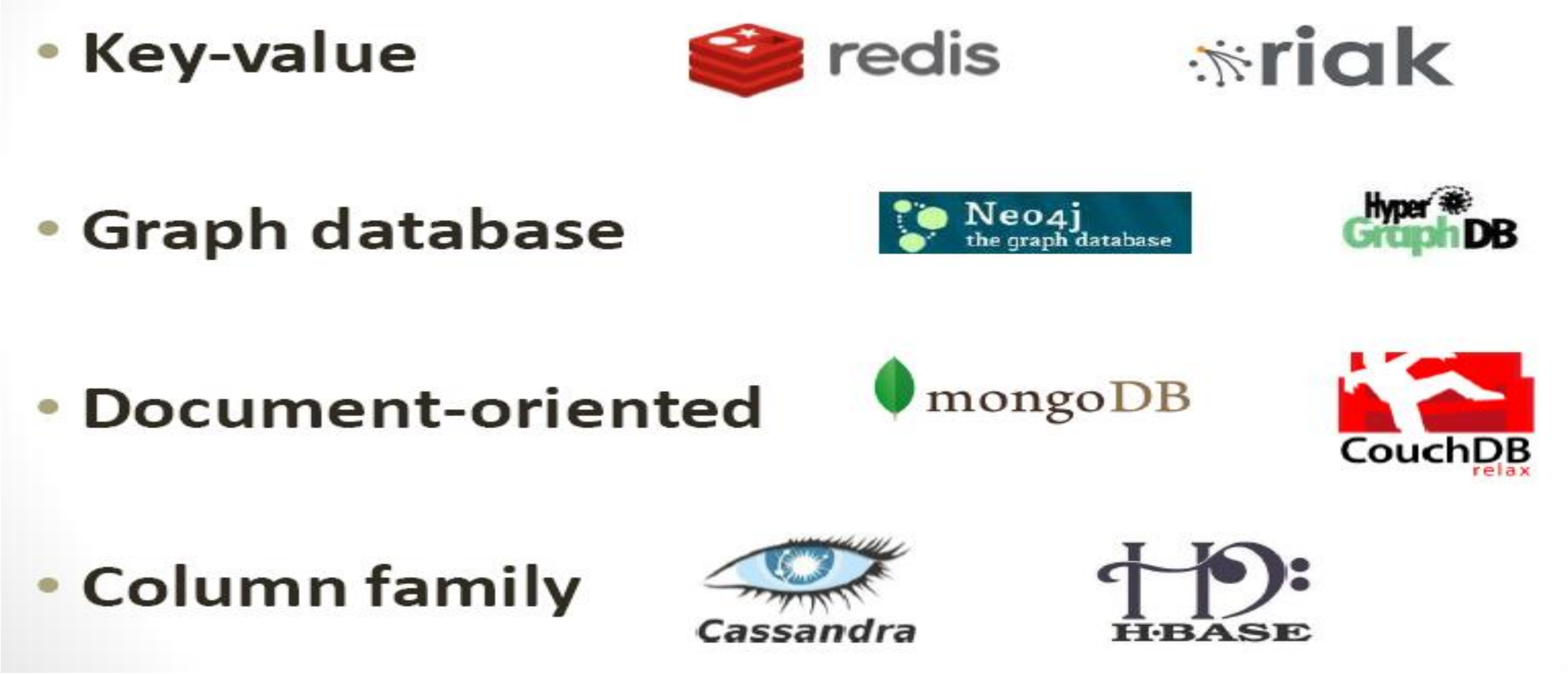
Available (CP)

Soft-state
(State of system may change over time)

Eventually consistent

(Asynchronous propagation)

NoSQL Types

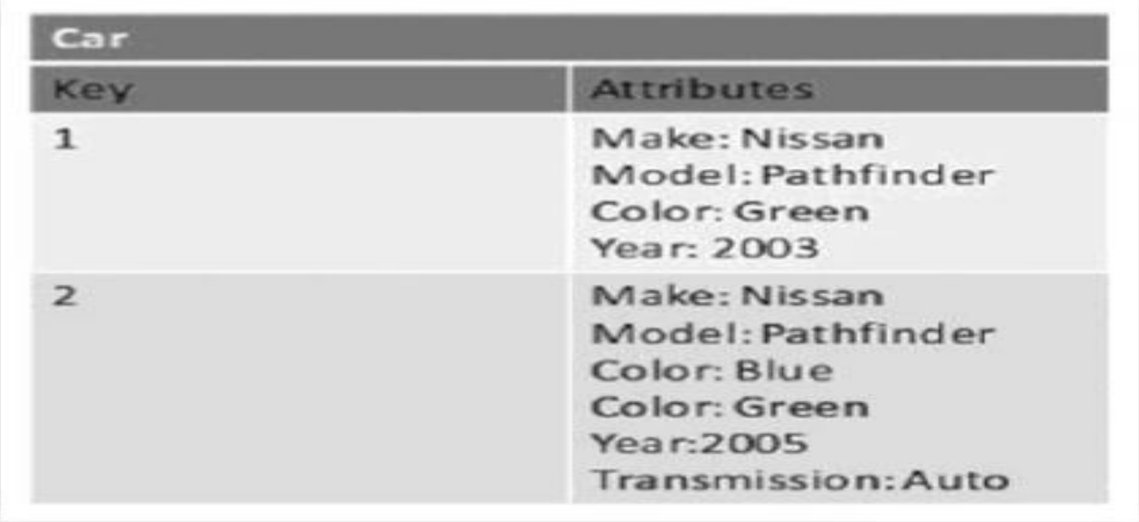


- Key-value
- Graph database
- Document-oriented
- Column family

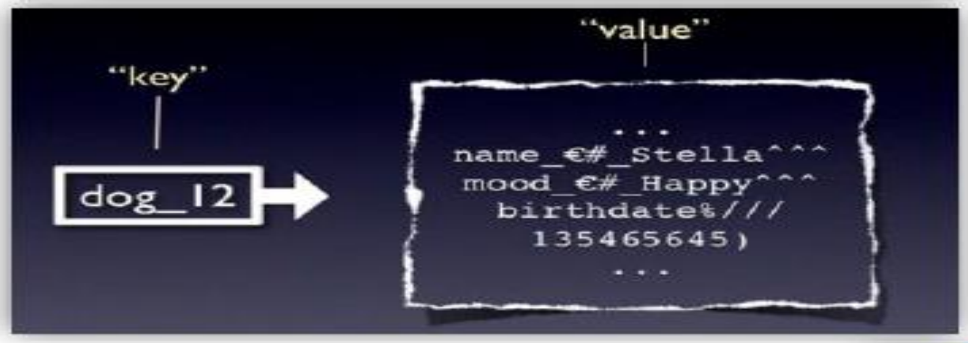
redis
riak
Neo4j
the graph database
HyperGraphDB
mongoDB
CouchDB
relax
Cassandra
H·BASE

Key-Value Data Model

- Simplest NOSQL databases
- The main idea is the use of a hash table
- Access data (values) by strings called keys
- Data has no required format data may have any format
- Data model: (key, value) pairs
- Basic Operations: Insert(key,value), Fetch(key), Update(key), Delete(key)



Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission:Auto

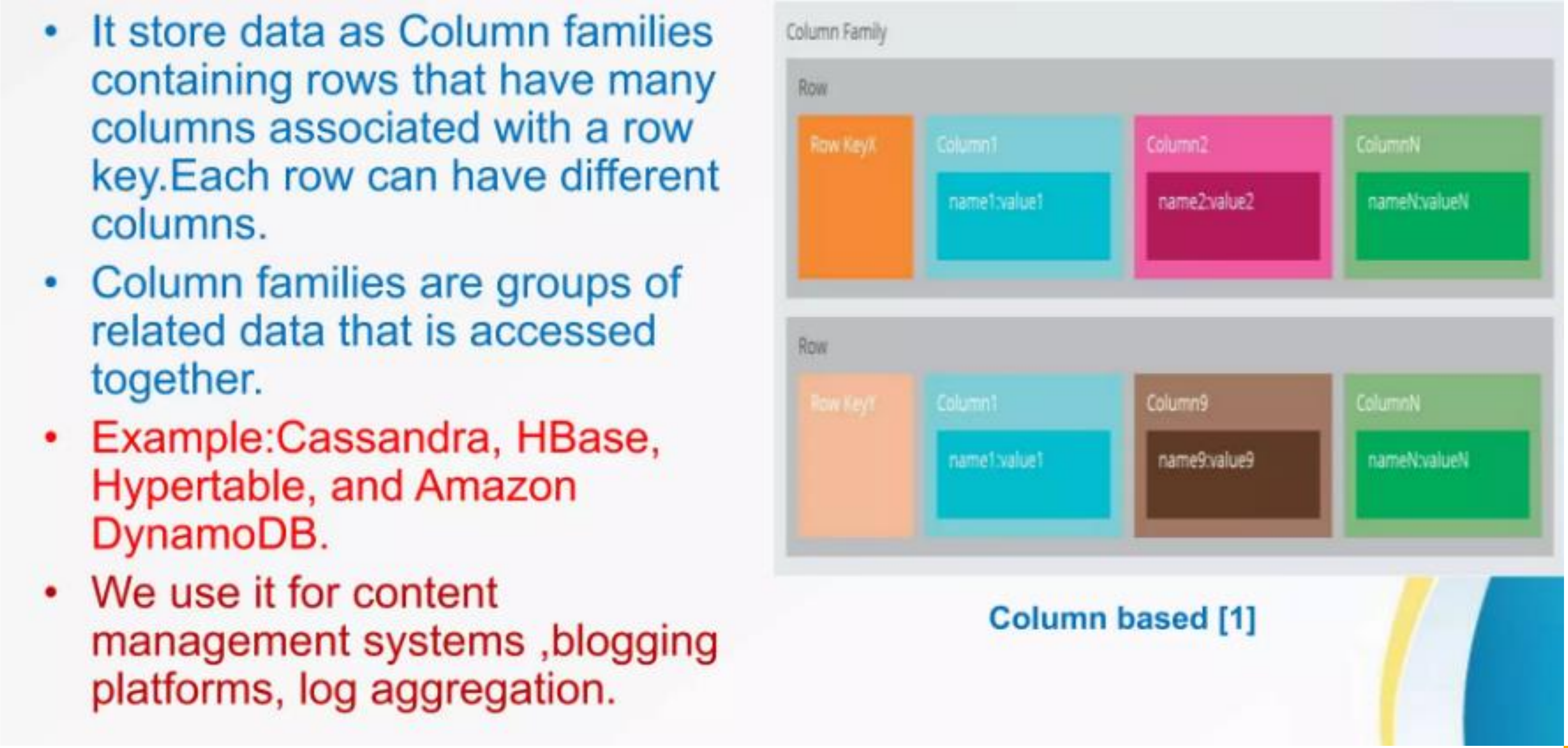


```
...
name_€#_Stella^^^
mood_€#_Happy^^^
birthdate%///
135465645)
...
```

amazon DynamoDB

- Example: Oracle NoSQL Database, Riak etc.
- We use it for: storing session information, user profiles, preferences , shopping cart data.

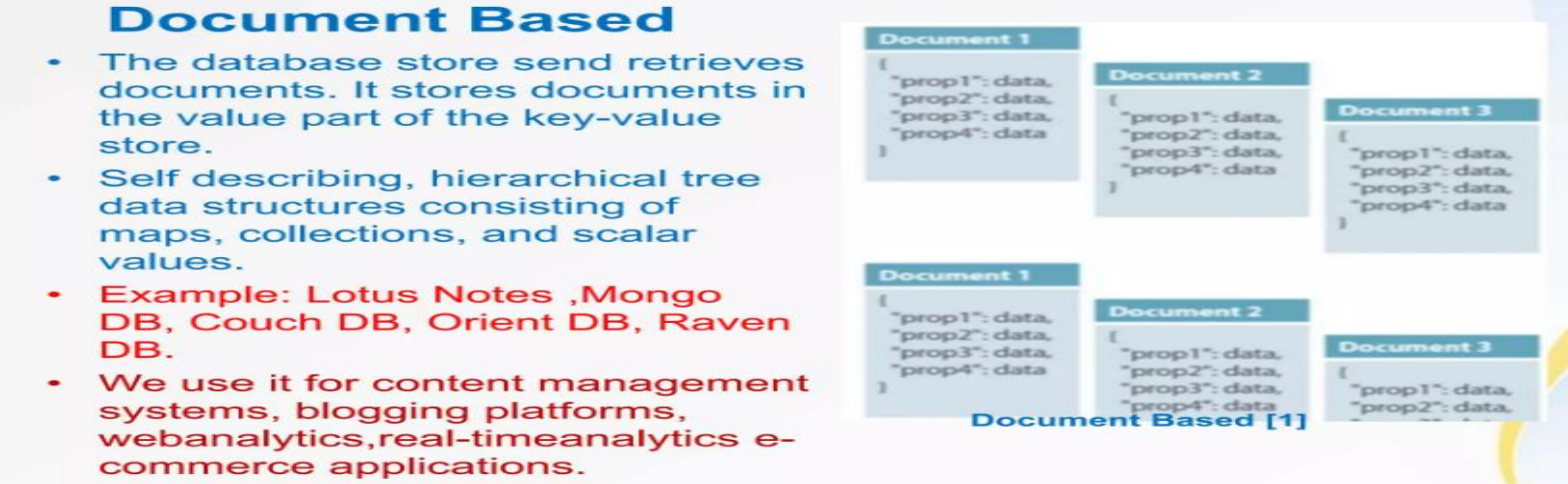
Column-Oriented



- It store data as Column families containing rows that have many columns associated with a row key.Each row can have different columns.
- Column families are groups of related data that is accessed together.
- Example:Cassandra, HBase, Hypertable, and Amazon DynamoDB.
- We use it for content management systems ,blogging platforms, log aggregation.

Column based [1]

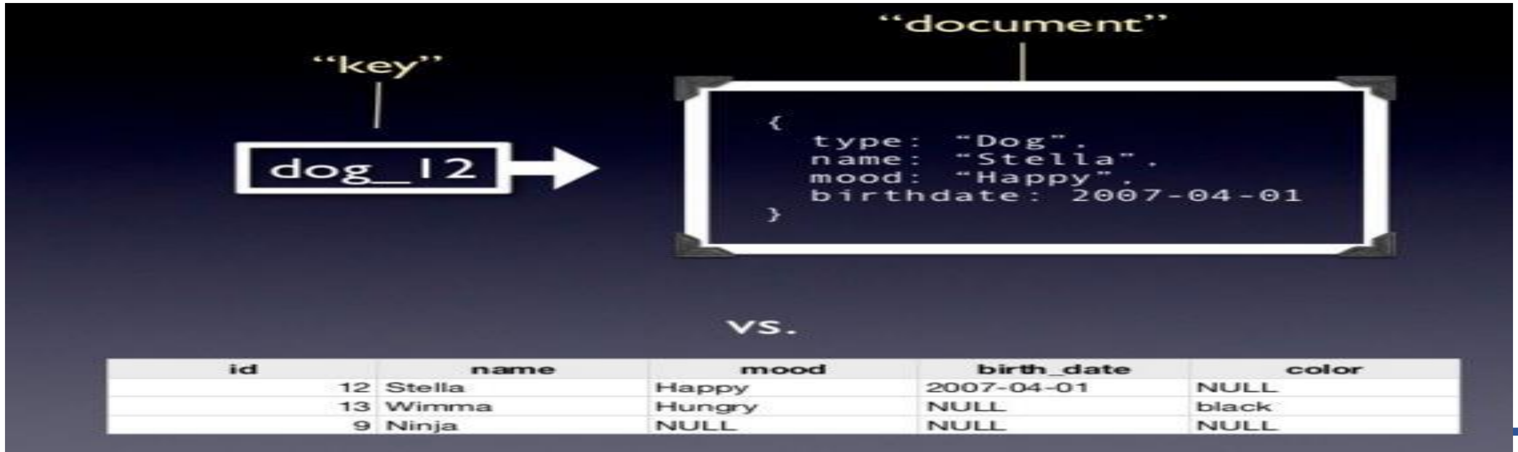
Document Oriented



Document Based

- The database store send retrieves documents. It stores documents in the value part of the key-value store.
- Self describing, hierarchical tree data structures consisting of maps, collections, and scalar values.
- Example: Lotus Notes ,Mongo DB, Couch DB, Orient DB, Raven DB.
- We use it for content management systems, blogging platforms, webanalytics,real-timeanalytics e-commerce applications.

Document Based [1]

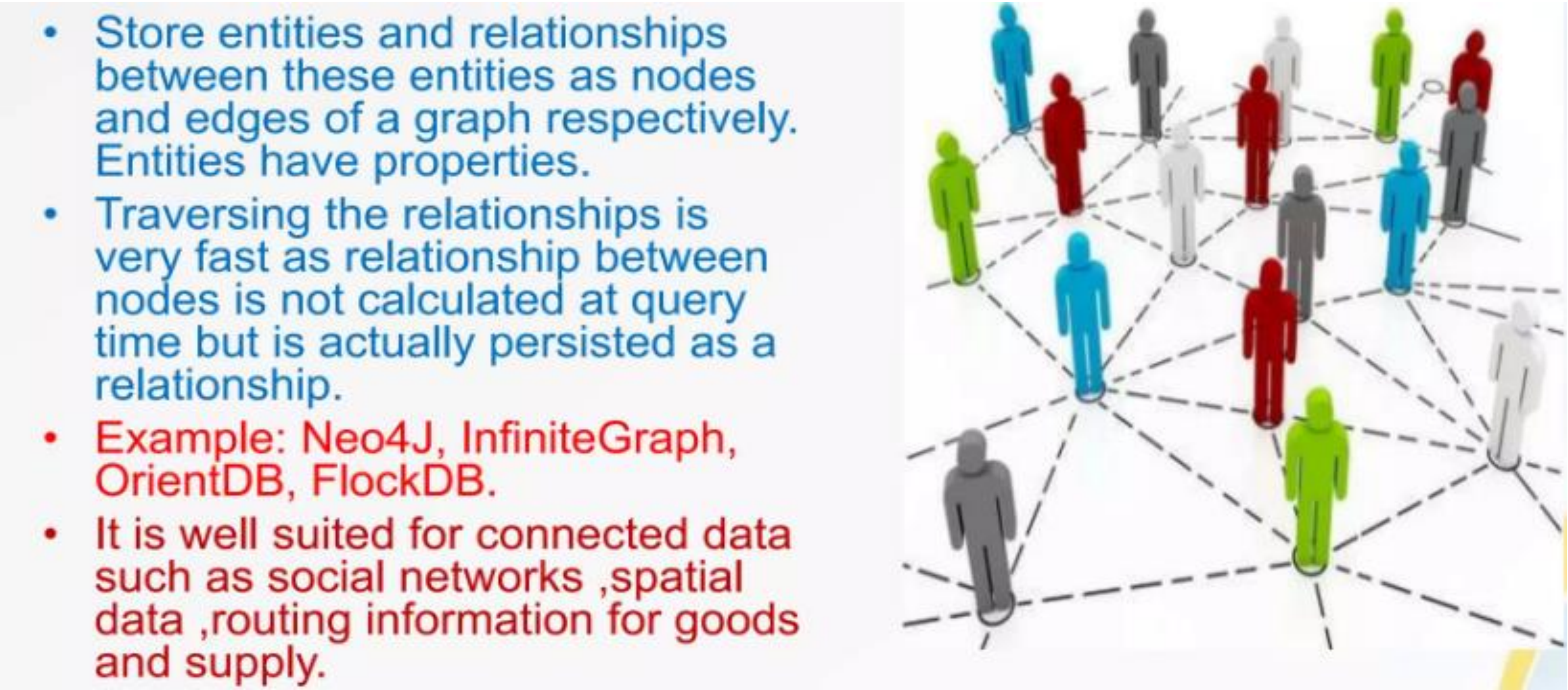


```
{
  type:  "Dog",
  name:  "Stella",
  mood:  "Happy",
  birthdate:  2007-04-01
}
```

vs.

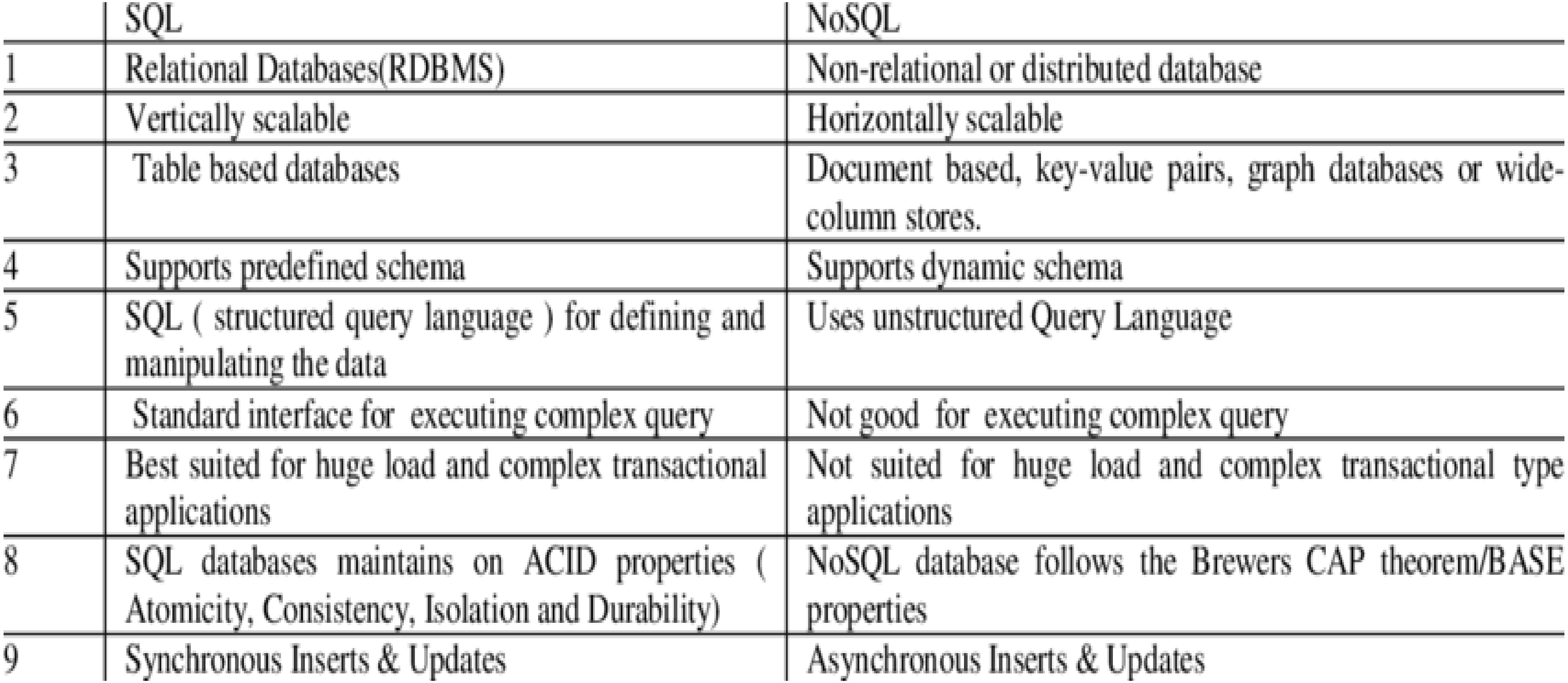
id	name	mood	birth_date	color
12	Stella	Happy	2007-04-01	NULL
13	Wimma	Hungry	NULL	black
9	Ninja	NULL	NULL	NULL

Graph Data Model



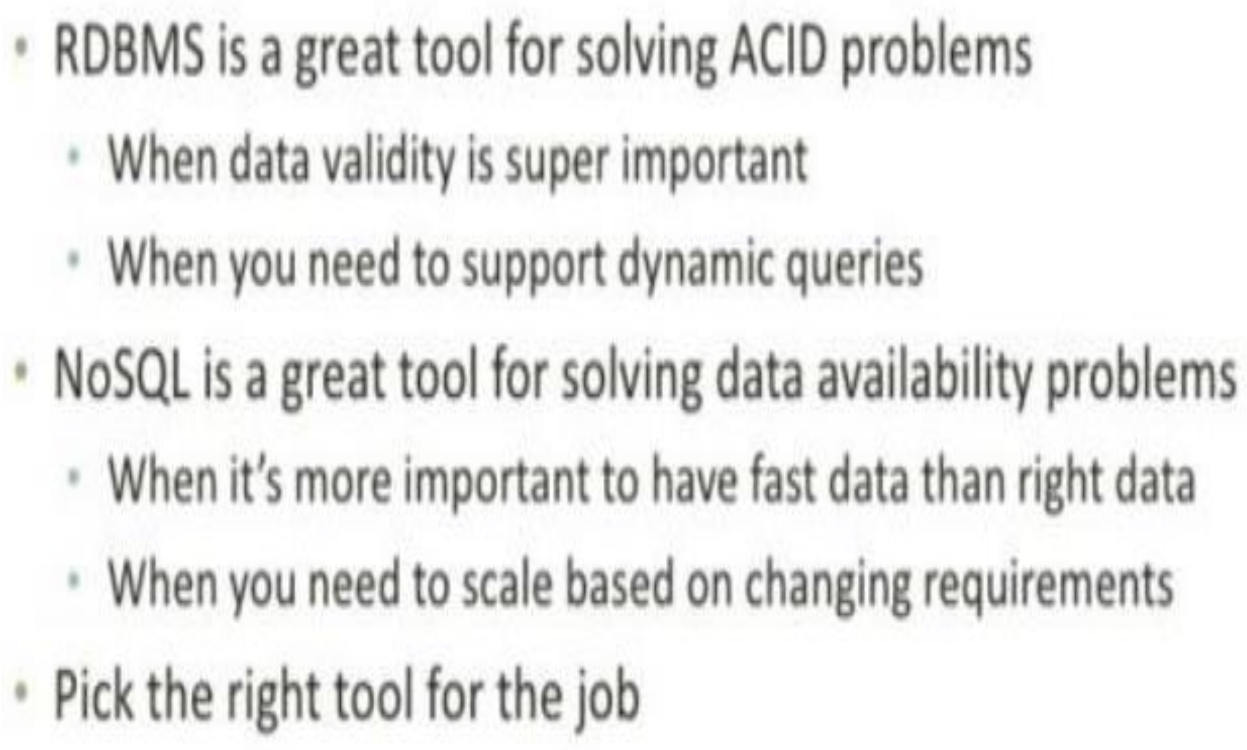
- Store entities and relationships between these entities as nodes and edges of a graph respectively. Entities have properties.
- Traversing the relationships is very fast as relationship between nodes is not calculated at query time but is actually persisted as a relationship.
- Example: Neo4J, InfiniteGraph, OrientDB, FlockDB.
- It is well suited for connected data such as social networks ,spatial data ,routing information for goods and supply.

SQL vs. NoSQL



	SQL	NoSQL
1	Relational Databases(RDBMS)	Non-relational or distributed database
2	Vertically scalable	Horizontally scalable
3	Table based databases	Document based, key-value pairs, graph databases or wide-column stores.
4	Supports predefined schema	Supports dynamic schema
5	SQL (structured query language) for defining and manipulating the data	Uses unstructured Query Language
6	Standard interface for executing complex query	Not good for executing complex query
7	Best suited for huge load and complex transactional applications	Not suited for huge load and complex transactional type applications
8	SQL databases maintains on ACID properties (Atomicity, Consistency, Isolation and Durability)	NoSQL database follows the Brewers CAP theorem/BASE properties
9	Synchronous Inserts & Updates	Asynchronous Inserts & Updates

Conclusion



- RDBMS is a great tool for solving ACID problems
  - When data validity is super important
  - When you need to support dynamic queries
- NoSQL is a great tool for solving data availability problems
  - When it's more important to have fast data than right data
  - When you need to scale based on changing requirements
- Pick the right tool for the job

Thank you