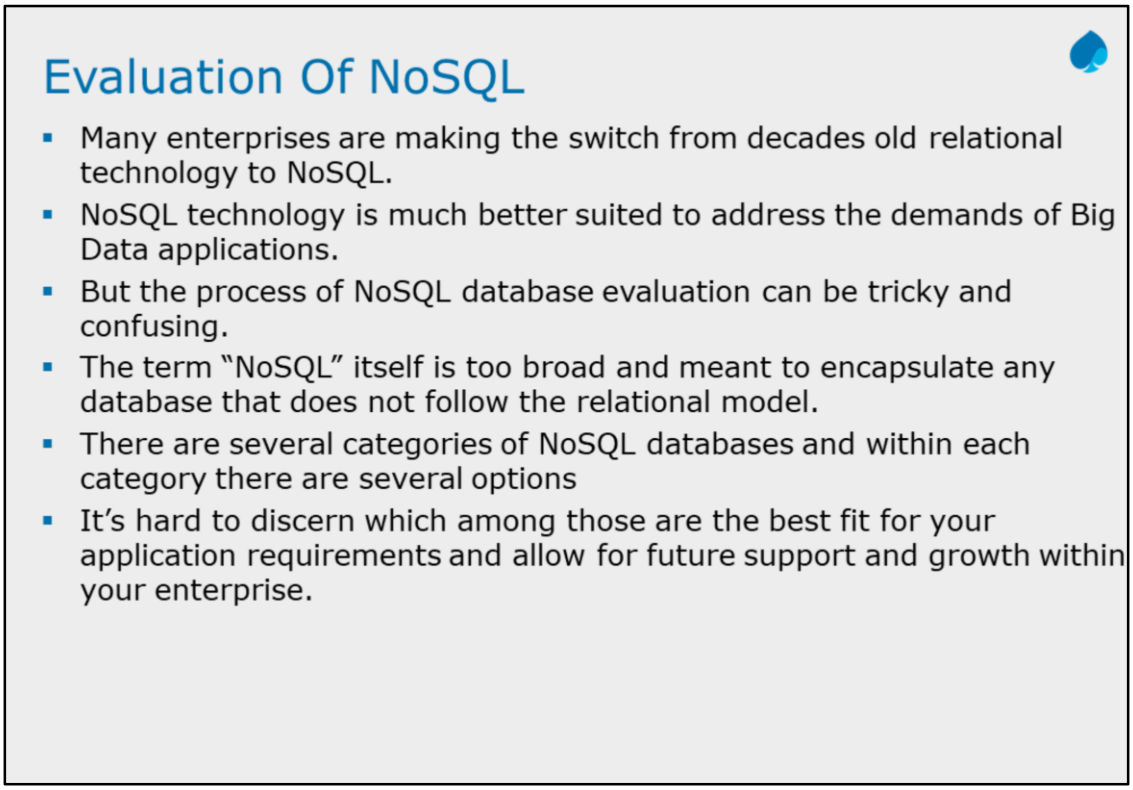
NO SQL Basics

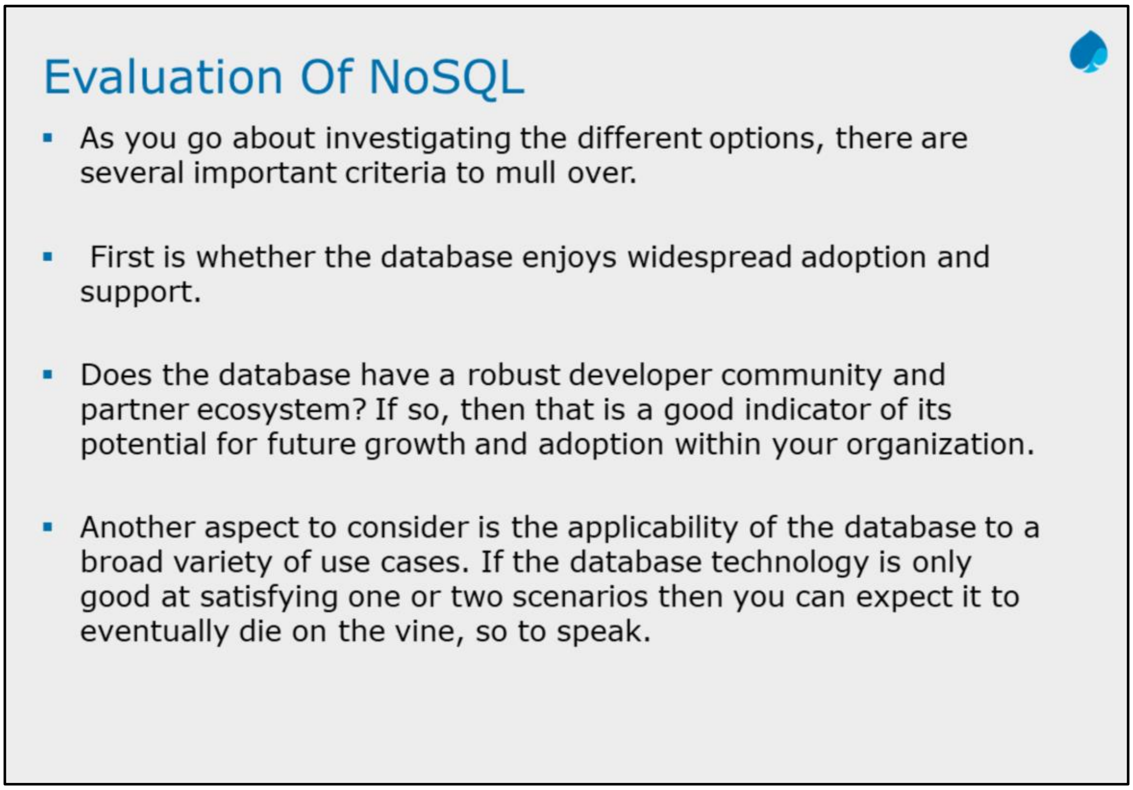
Lesson 01

Capgemini



Evaluation Of NoSQL

- Many enterprises are making the switch from decades old relational technology to NoSQL.
- NoSQL technology is much better suited to address the demands of Big Data applications.
- But the process of NoSQL database evaluation can be tricky and confusing.
- The term "NoSQL" itself is too broad and meant to encapsulate any database that does not follow the relational model.
- There are several categories of NoSQL databases and within each category there are several options
- It's hard to discern which among those are the best fit for your application requirements and allow for future support and growth within your enterprise.



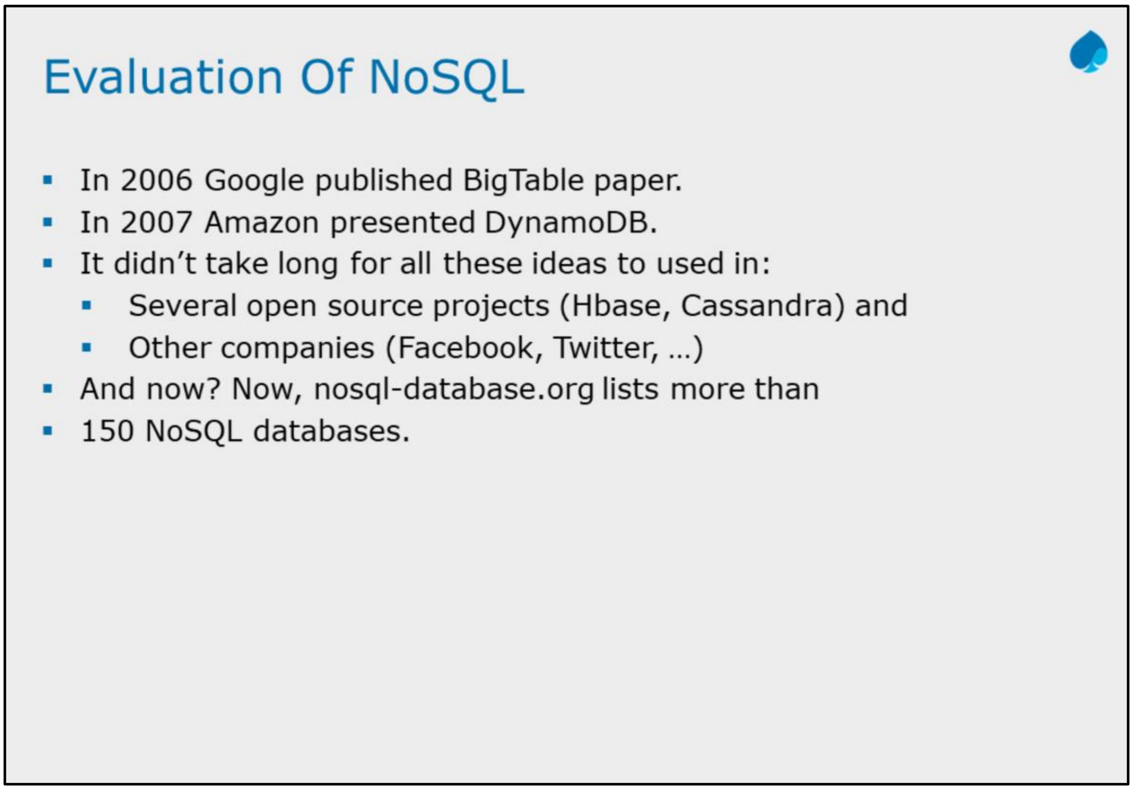
Evaluation Of NoSQL

- As you go about investigating the different options, there are several important criteria to mull over.
- First is whether the database enjoys widespread adoption and support.
- Does the database have a robust developer community and partner ecosystem? If so, then that is a good indicator of its potential for future growth and adoption within your organization.
- Another aspect to consider is the applicability of the database to a broad variety of use cases. If the database technology is only good at satisfying one or two scenarios then you can expect it to eventually die on the vine, so to speak.

Based on just these two criteria alone, MongoDB is the NoSQL database to consider for your modern, Big Data applications.

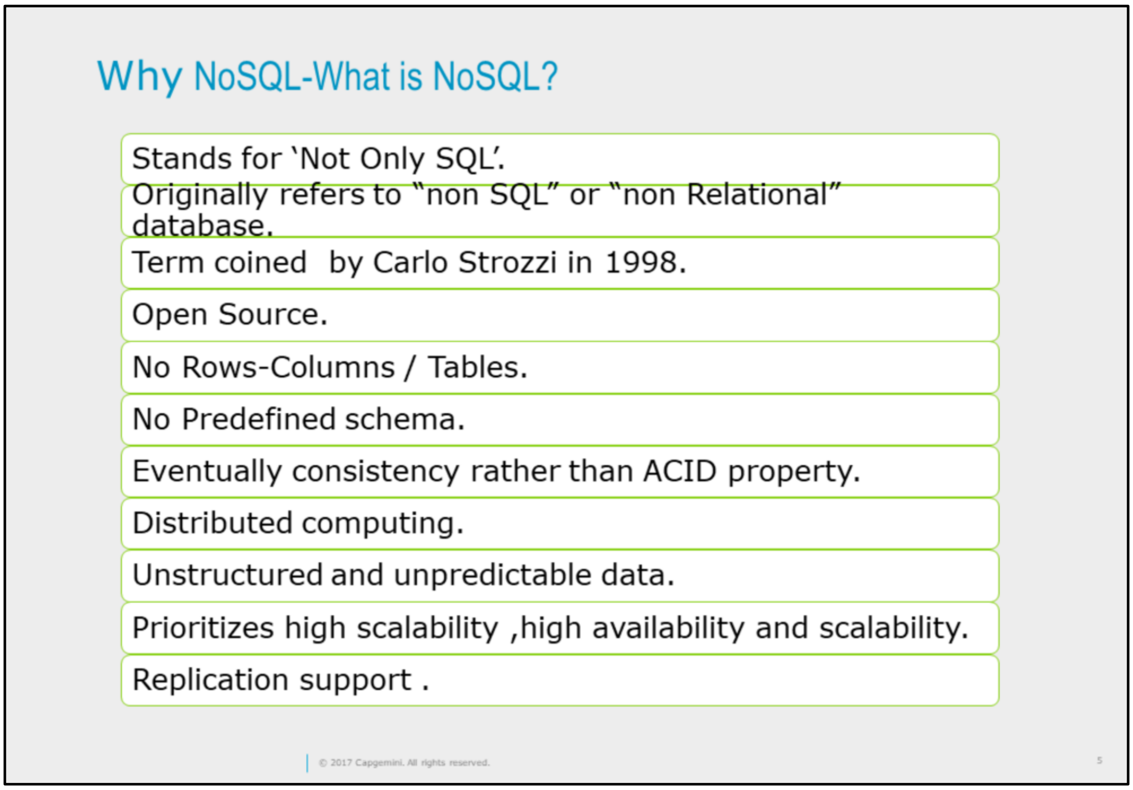
The most popular database with 10 million downloads and counting, MongoDB has a thriving developer community with thousands of certified professionals, and it consistently ranks as the most popular.

NoSQL database according to DB-Engines' monthly rankings. Also, as a general purpose database you can successfully employ MongoDB to address many different use cases.



Evaluation Of NoSQL

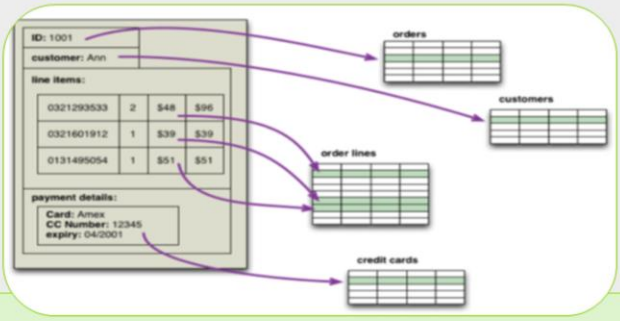
- In 2006 Google published BigTable paper.
- In 2007 Amazon presented DynamoDB.
- It didn’t take long for all these ideas to used in:
  - Several open source projects (Hbase, Cassandra) and
  - Other companies (Facebook, Twitter, …)
- And now? Now, nosql-database.org lists more than
- 150 NoSQL databases.



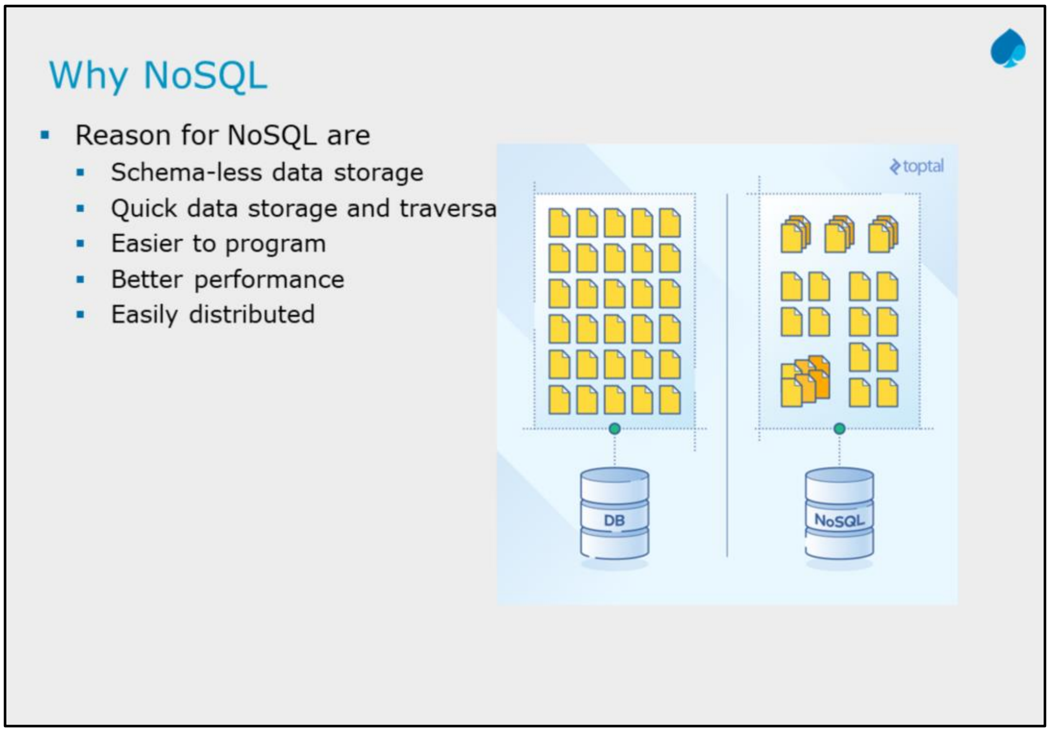
Why NoSQL-What is NoSQL?

- Stands for 'Not Only SQL'.
- Originally refers to "non SQL" or "non Relational" database.
- Term coined by Carlo Strozzi in 1998.
- Open Source.
- No Rows-Columns / Tables.
- No Predefined schema.
- Eventually consistency rather than ACID property.
- Distributed computing.
- Unstructured and unpredictable data.
- Prioritizes high scalability ,high availability and scalability.
- Replication support .

Why NoSQL

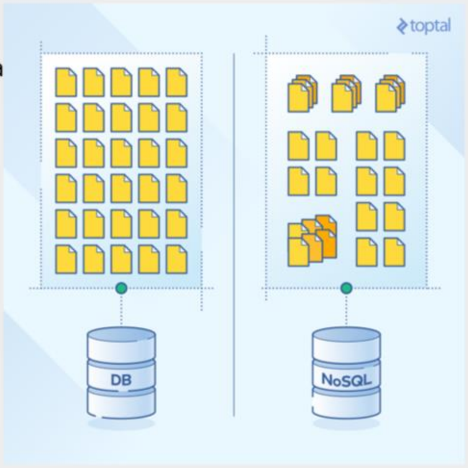


- Handles Schema Changes Well (easy development)
- Solves Impedance Mismatch problem
- Rise of JSON
  - python module: simplejson



Why NoSQL

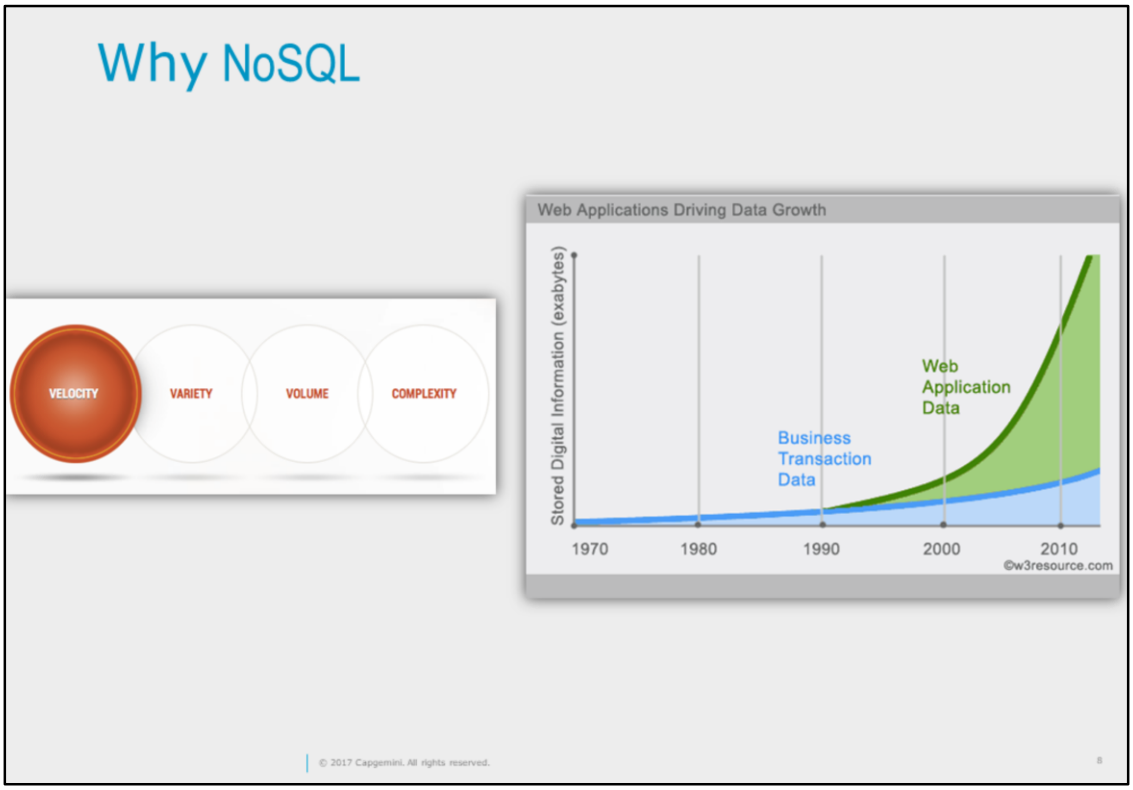
- Reason for NoSQL are
  - Schema-less data storage
  - Quick data storage and traversa
  - Easier to program
  - Better performance
  - Easily distributed



One key difference between NoSQL databases and traditional relational databases is the fact that NoSQL is a form of *unstructured storage*.

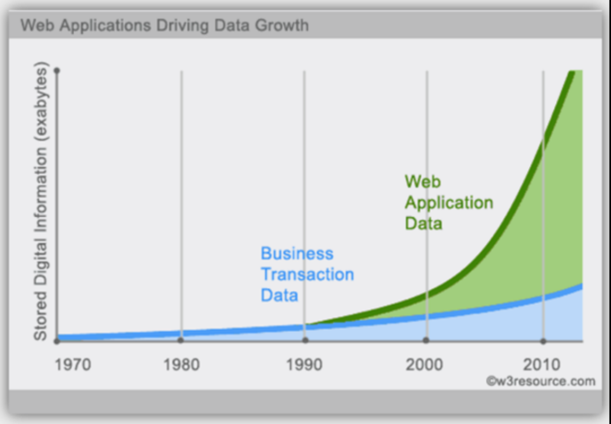
With the explosion of social media, user driven content has grown rapidly and has increased the volume and type of data that is produced, managed, analyzed, and archived. In addition, new sources of data, such as sensors, Global Positioning Systems or GPS, automated trackers, and other monitoring systems generate huge volumes of data on a regular basis.

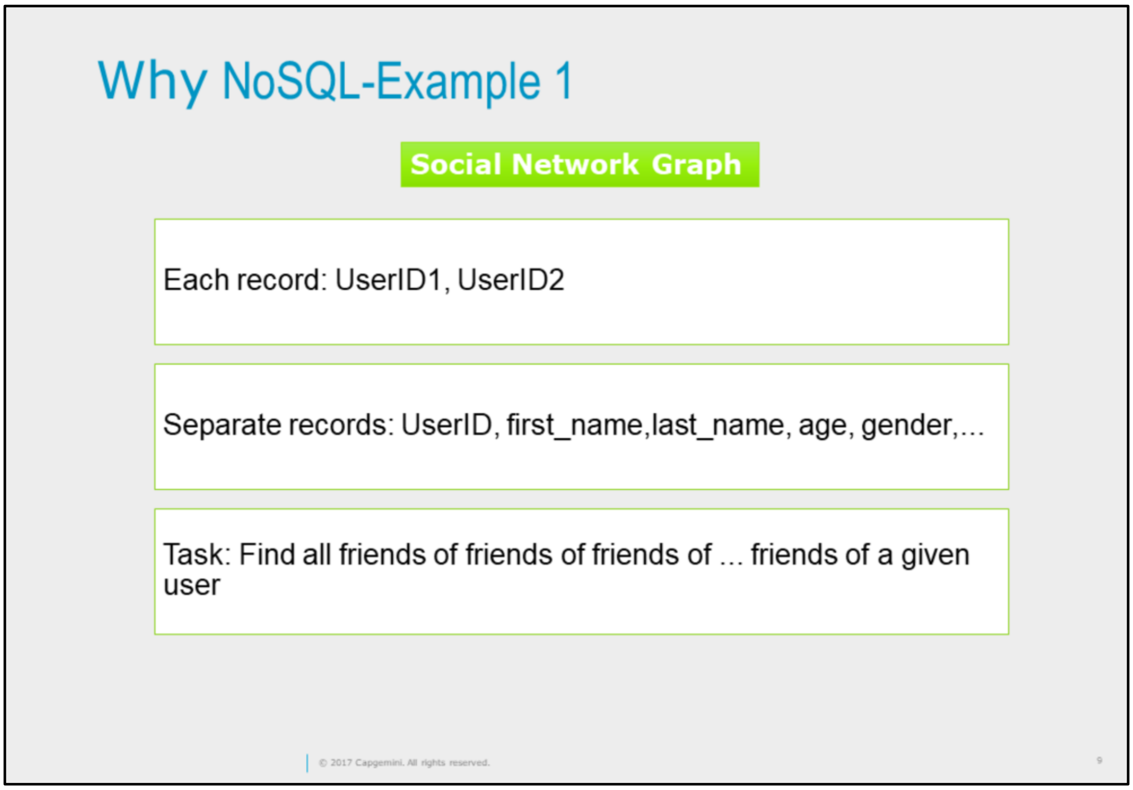
To resolve the problems related to large-volume and semi-structured data, a class of new database products have emerged. These new classes of database products consist of column-based data stores, key/value pair databases, and document databases. Together, these are called NoSQL. The NoSQL database consists of diverse products with each product having unique sets of features and value propositions.



Why NoSQL

VELOCITY
VARIETY
VOLUME
COMPLEXITY


Web Applications Driving Data Growth
Stored Digital Information (exabytes)
Web Application Data
Business Transaction Data
1970
1980
1990
2000
2010
©w3resource.com



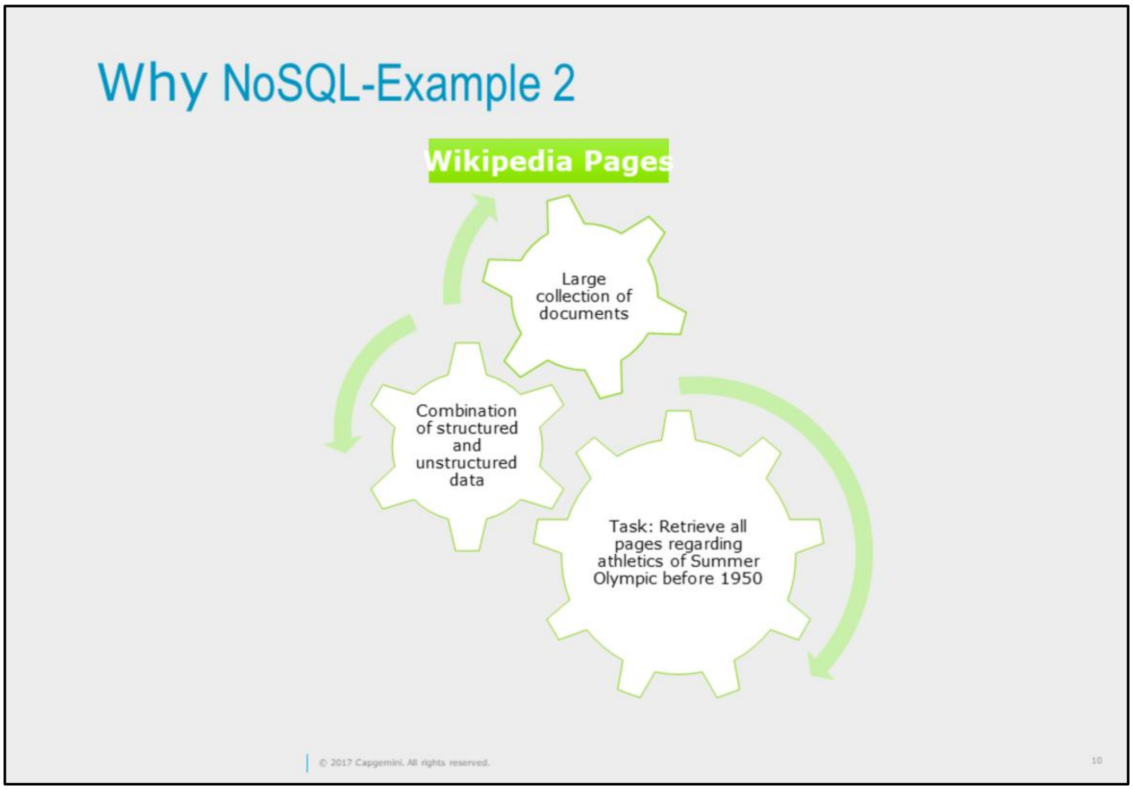
Why NoSQL-Example 1

Social Network Graph

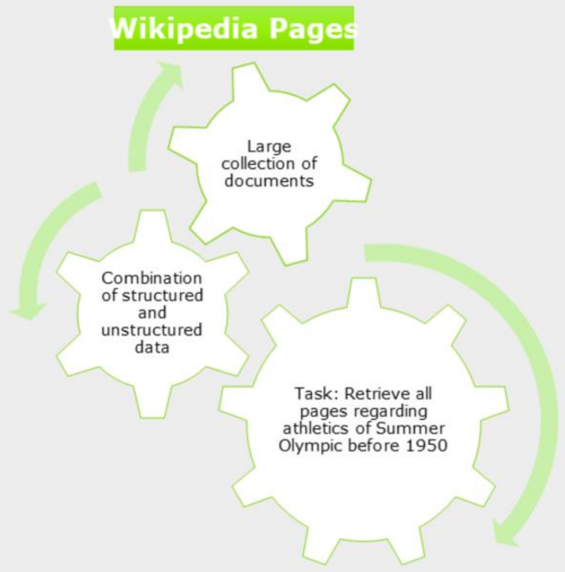
Each record: UserID1, UserID2

Separate records: UserID, first_name,last_name, age, gender,...

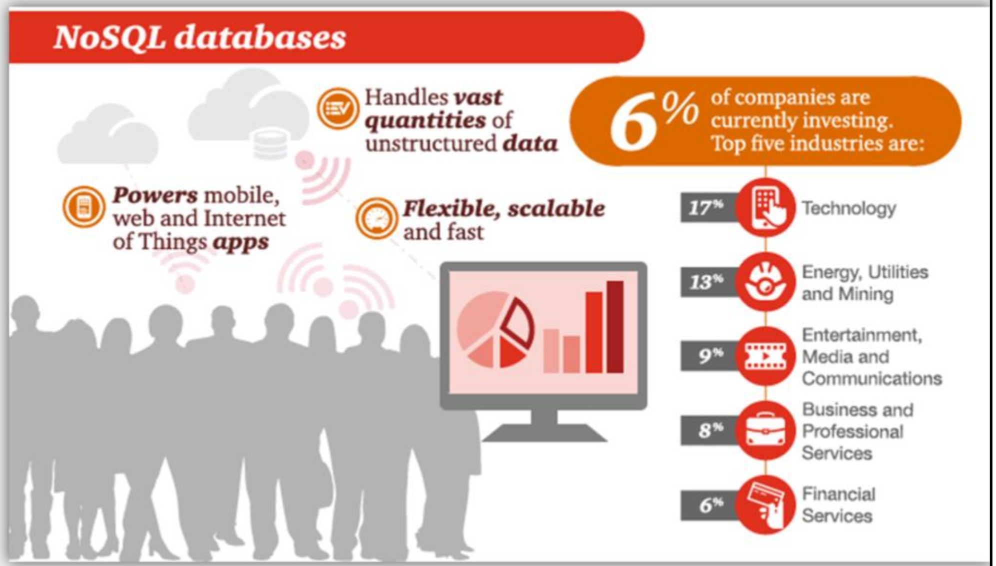
Task: Find all friends of friends of friends of ... friends of a given user

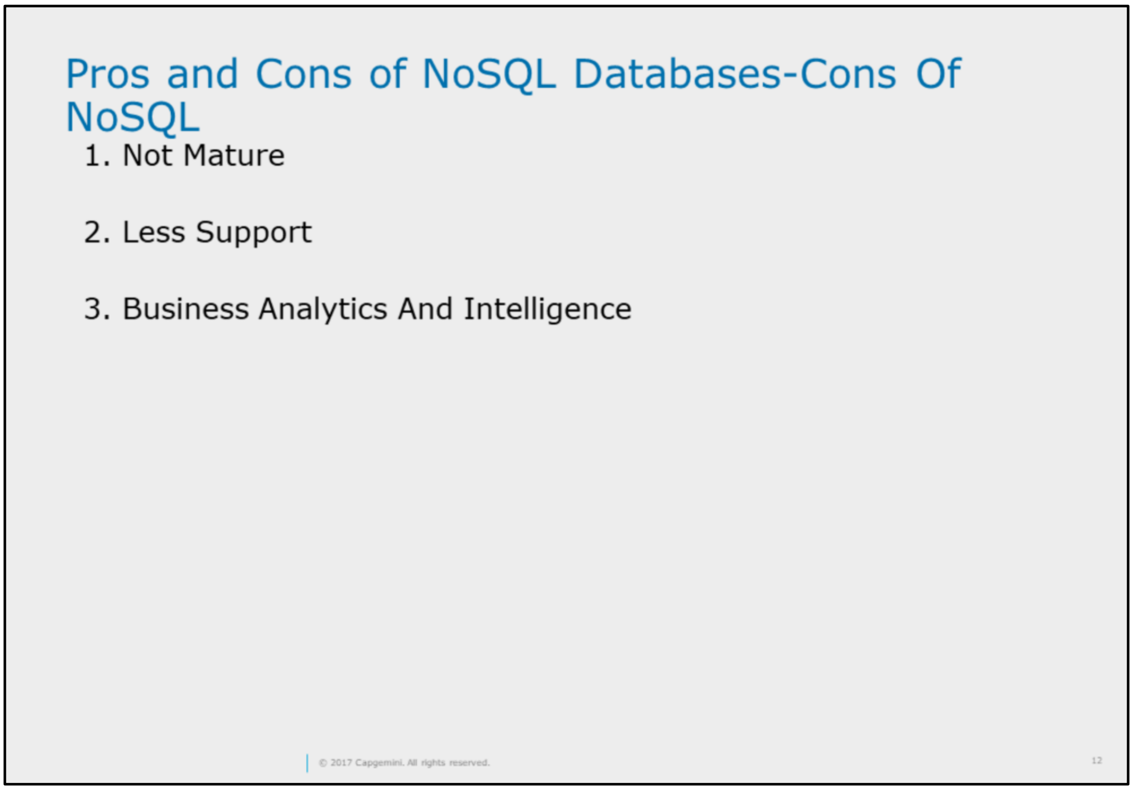


Why NoSQL-Example 2


Wikipedia Pages
Large collection of documents
Combination of structured and unstructured data
Task: Retrieve all pages regarding athletics of Summer Olympic before 1950

Why NoSQL - NOSQL Market



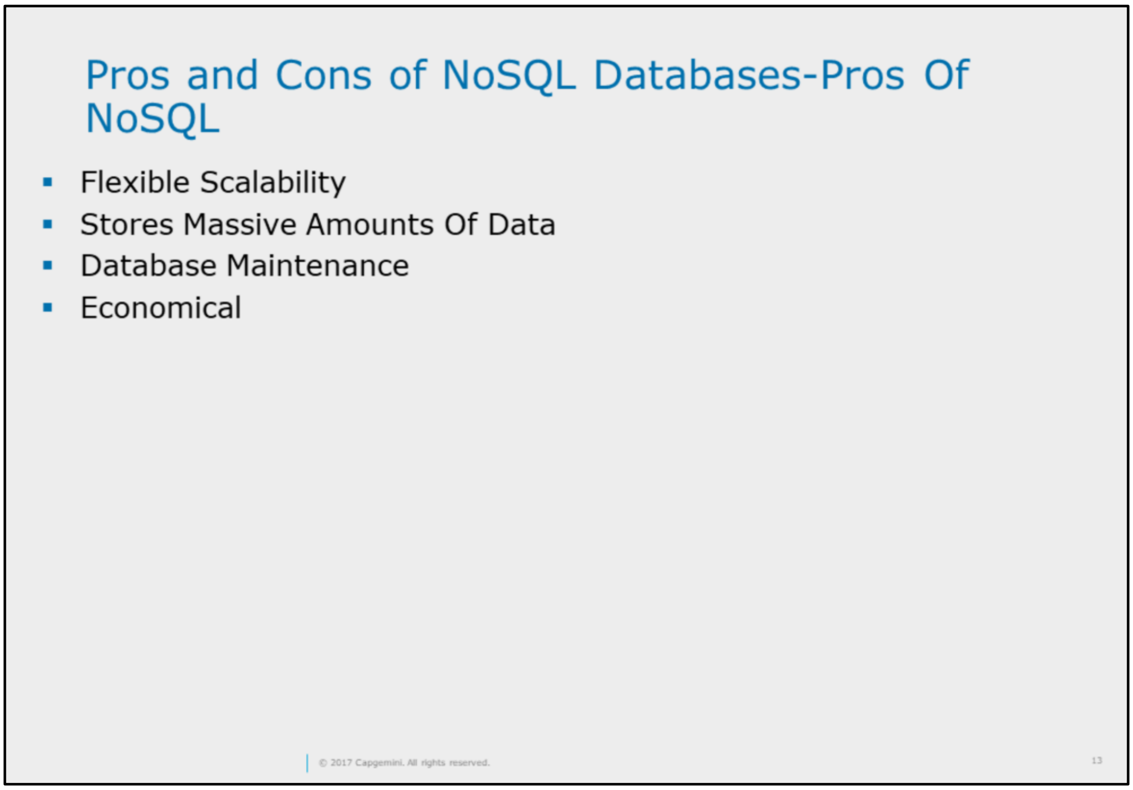

Pros and Cons of NoSQL Databases-Cons Of NoSQL

1. Not Mature

2. Less Support

3. Business Analytics And Intelligence

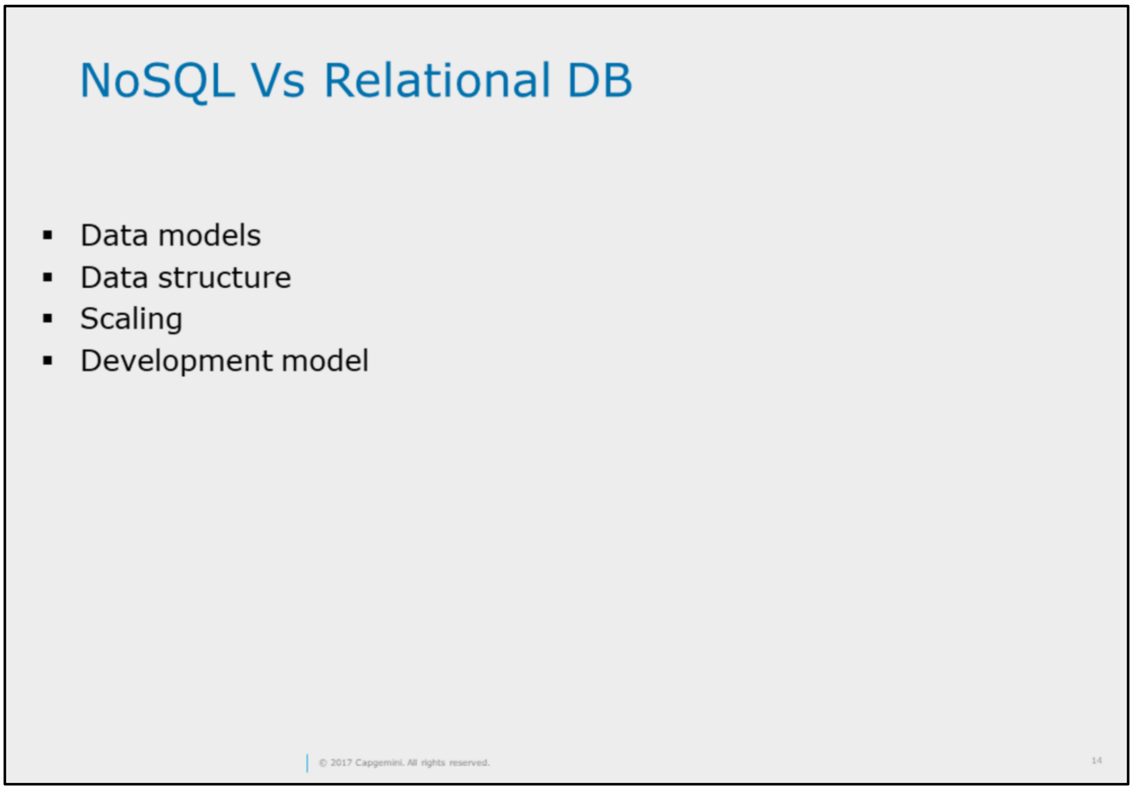
1. Not Mature
Rational models have been around for some time now compared to NoSQL models and as a result they have grown to be more functional and stable systems over the years.
2. Less Support
Every business should be reassured that in case a key function in their database system fails, they will have unlimited competent support any time. All rational model vendors have gone the extra mile to provide this assurance and made it sure that their support is available 24 hours which is not a step yet guaranteed by NoSQL vendors.
3. Business Analytics And Intelligence
NoSQL models were created because of the modern-day web 2.0 web applications in mind. And because of this, most NoSQL features are focused to meeting these demands ignoring the demands of apps made without these characteristics hence end up offering fewer analytic features for normal web apps.
Any businesses looking to implement NoSQL model needs to do it with caution, remembering the above mentioned pros and cons they posse in contrast to their rational opposites.


Pros and Cons of NoSQL Databases-Pros Of NoSQL

- Flexible Scalability
- Stores Massive Amounts Of Data
- Database Maintenance
- Economical

13

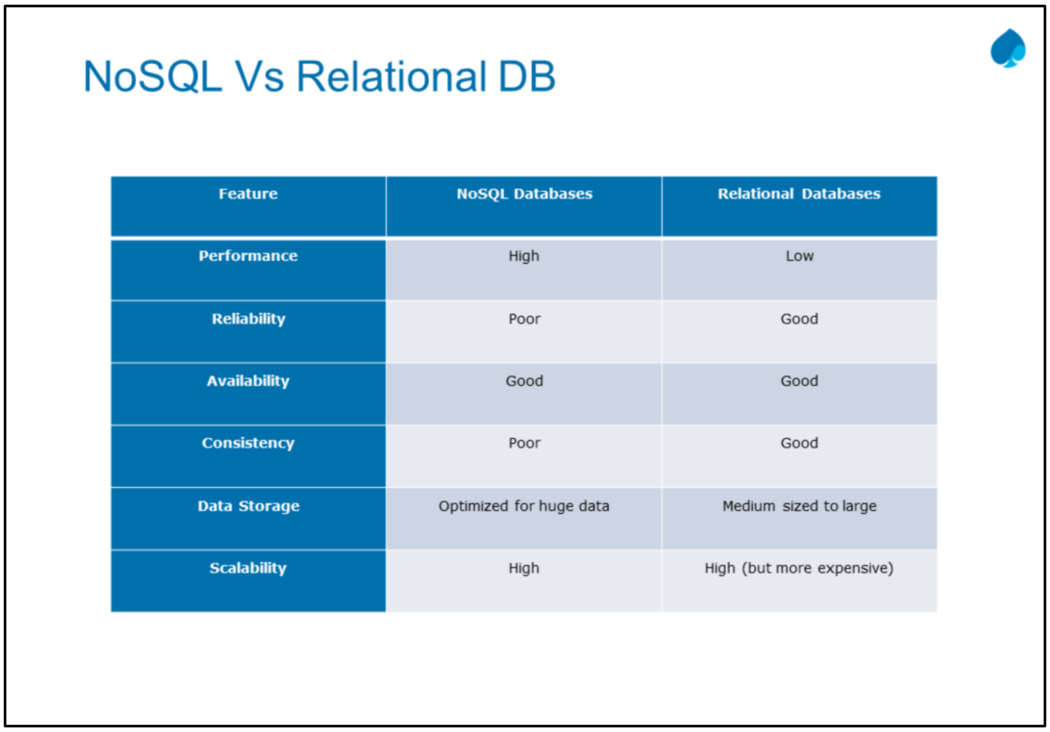
1.Flexible Scalability
Unlike rational database management model that is difficult to scale out when it come to commodity clusters NoSQL models make use of new nodes which makes them transparent for expansion. The model is designed to be used even with low cost hardware's. In this current world where outward scalability is replacing upwards scalability, NoSQL models are the better option.
2. Stores Massive Amounts Of Data
Given the fact that transaction rates are rising due to recognition, huge volumes of data need to be stored. While rational models have grown to meet this need it is illogical to use such models to store such large volumes of data. However these volumes can easily be handled by NoSQL models
3. Database Maintenance
The best rational models need the service of an expert to design, install and maintain. However, NoSQL models need much less expert management as it already has auto repair and data distribution capabilities, fewer administration and turning requirements as well as simplified data designs.
4. Economical
Rational models require expensive proprietary servers and storage systems whereas NoSQL models are easy and cheap to install. This means that more data can be processed and stored at a very minimal cost.



NoSQL Vs Relational DB

- Data models
- Data structure
- Scaling
- Development model

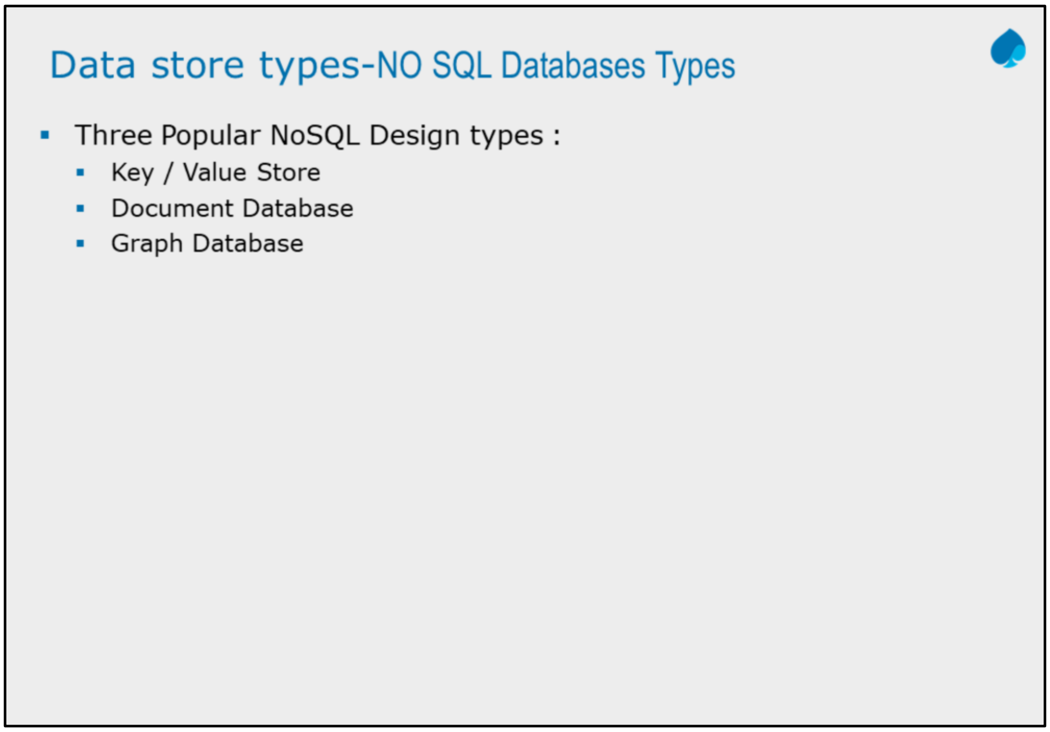
NoSQL databases differ from older, relational technology in four main areas:
Data models: A NoSQL database lets you build an application without having to define the schema first unlike relational databases which make you define your schema before you can add any data to the system. No predefined schema makes NoSQL databases much easier to update as your data and requirements change.
Data structure: Relational databases were built in an era where data was fairly structured and clearly defined by their relationships. NoSQL databases are designed to handle unstructured data (e.g., texts, social media posts, video, email) which makes up much of the data that exists today.
Scaling: It's much cheaper to scale a NoSQL database than a relational database because you can add capacity by scaling out over cheap, commodity servers. Relational databases, on the other hand, require a single server to host your entire database. To scale, you need to buy a bigger, more expensive server.
Development model: NoSQL databases are open source whereas relational databases typically are closed source with licensing fees baked into the use of their software. With NoSQL, you can get started on a project without any heavy investments in software fees upfront.



NoSQL Vs Relational DB

Feature	NoSQL Databases	Relational Databases
Performance	High	Low
Reliability	Poor	Good
Availability	Good	Good
Consistency	Poor	Good
Data Storage	Optimized for huge data	Medium sized to large
Scalability	High	High (but more expensive)

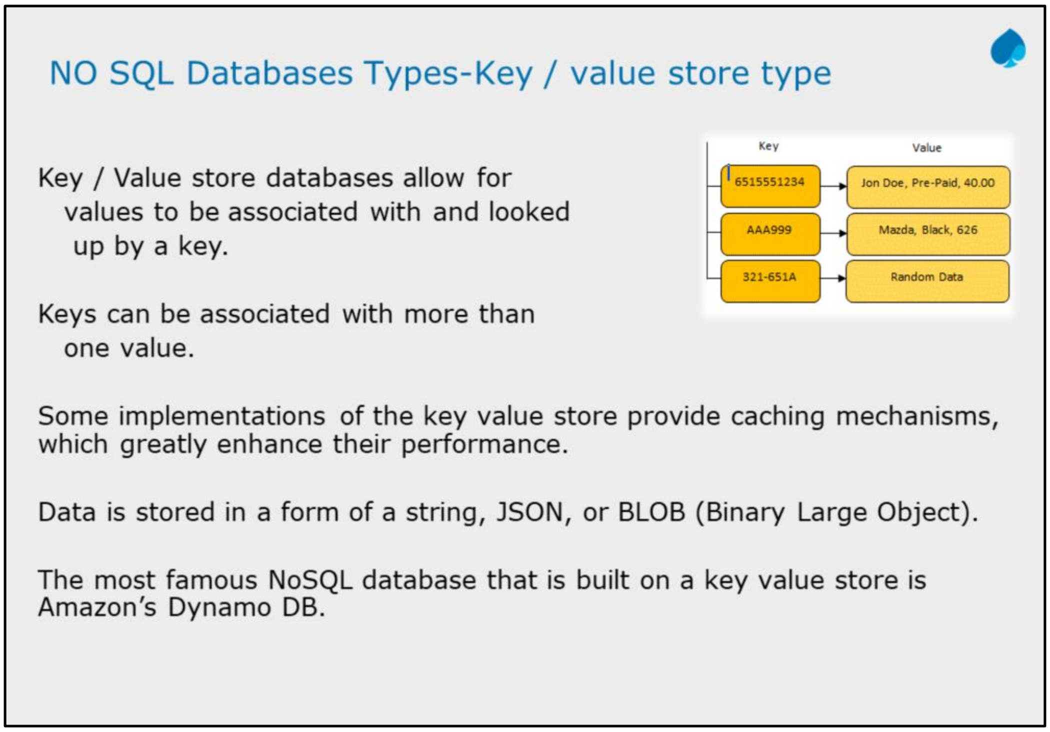
Most NoSQL databases do not support *reliability features(ACID)* which are natively supported by relational database System.
This also means that NoSQL databases, which don’t support those features, trade consistency for performance and scalability.



Data store types-NO SQL Databases Types

- Three Popular NoSQL Design types :
  - Key / Value Store
  - Document Database
  - Graph Database

Most NoSQL databases do not support *reliability features(ACID)* which are natively supported by relational database System.
This also means that NoSQL databases, which don't support those features, trade consistency for performance and scalability.



NO SQL Databases Types-Key / value store type

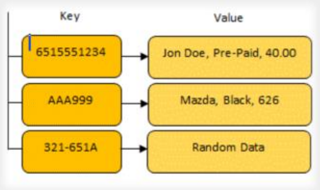
Key / Value store databases allow for values to be associated with and looked up by a key.

Keys can be associated with more than one value.

Some implementations of the key value store provide caching mechanisms, which greatly enhance their performance.

Data is stored in a form of a string, JSON, or BLOB (Binary Large Object).

The most famous NoSQL database that is built on a key value store is Amazon's Dynamo DB.

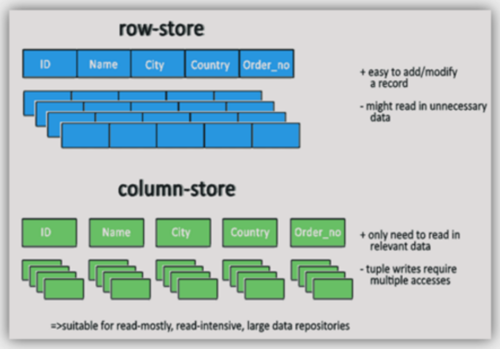


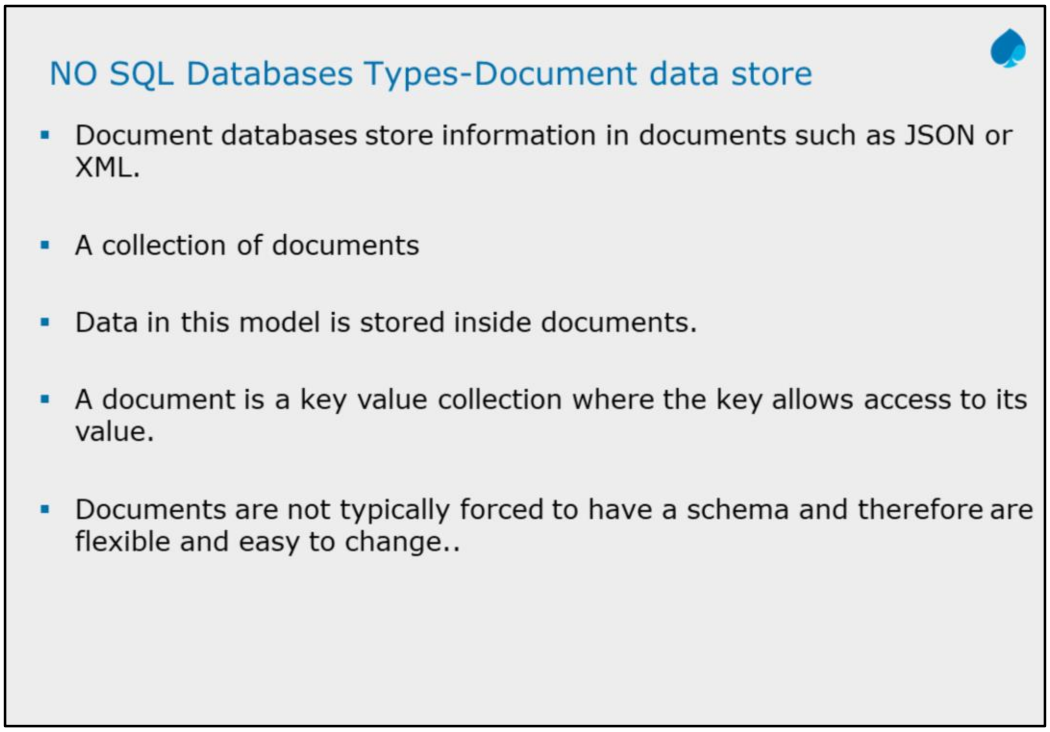
Most NoSQL databases do not support *reliability features(ACID)* which are natively supported by relational database System.

This also means that NoSQL databases, which don't support those features, trade consistency for performance and scalability.

NO SQL Databases Types-Key / value store type

Key Value Stores

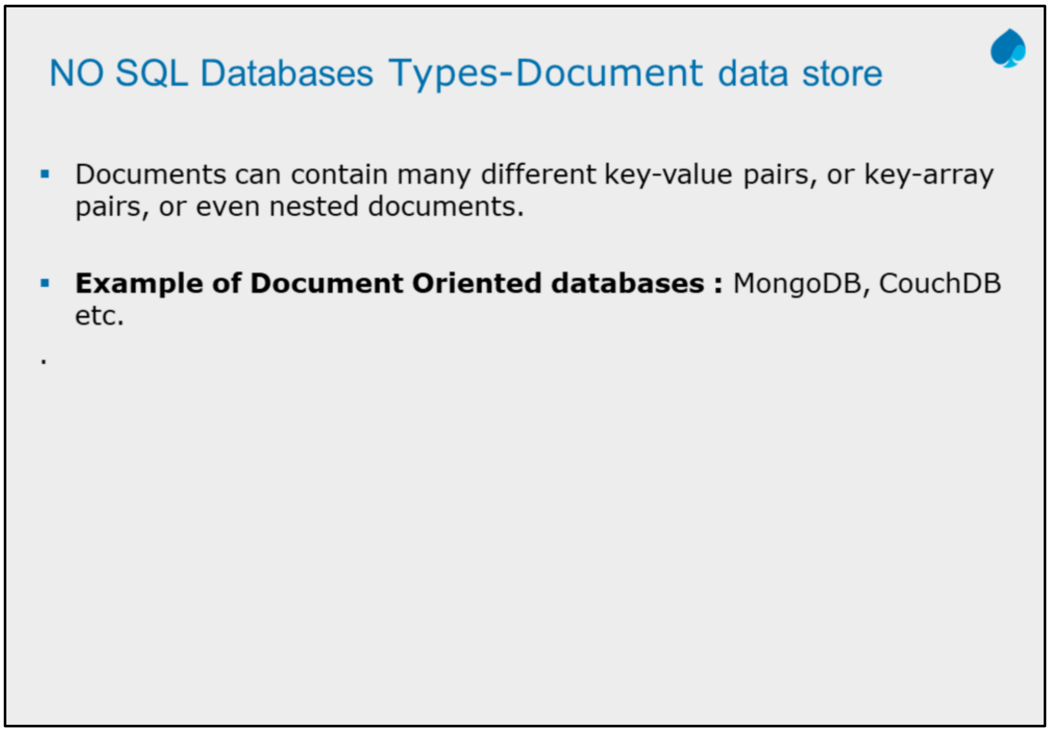

row-store
ID
Name
City
Country
Order_no
+ easy to add/modify a record
- might read in unnecessary data
column-store
ID
Name
City
Country
Order_no
+ only need to read in relevant data
- tuple writes require multiple accesses
=>suitable for read-mostly, read-intensive, large data repositories



NO SQL Databases Types-Document data store

- Document databases store information in documents such as JSON or XML.
- A collection of documents
- Data in this model is stored inside documents.
- A document is a key value collection where the key allows access to its value.
- Documents are not typically forced to have a schema and therefore are flexible and easy to change..

Most NoSQL databases do not support *reliability features(ACID)* which are natively supported by relational database System.
This also means that NoSQL databases, which don’t support those features, trade consistency for performance and scalability.



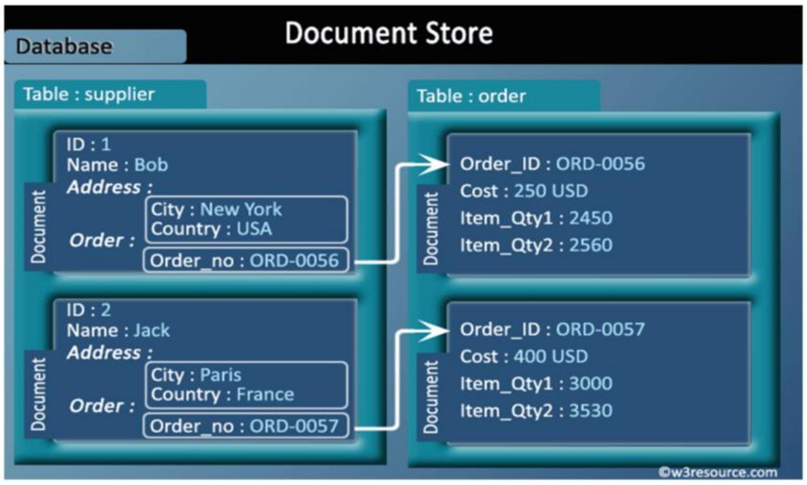
NO SQL Databases Types-Document data store

- Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.
- **Example of Document Oriented databases :** MongoDB, CouchDB etc.

.

Most NoSQL databases do not support *reliability features(ACID)* which are natively supported by relational database System.
This also means that NoSQL databases, which don't support those features, trade consistency for performance and scalability.

NO SQL Databases Types-Document data store

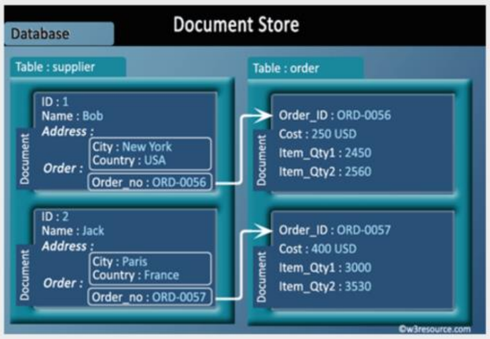


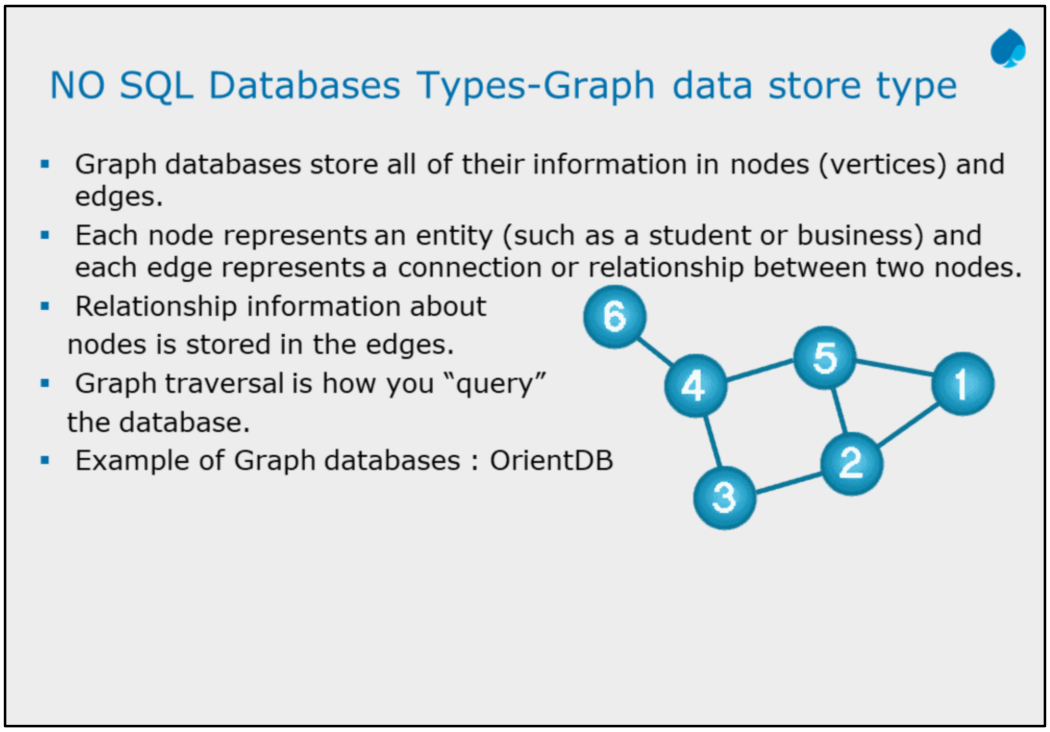
Most NoSQL databases do not support *reliability features(ACID)* which are natively supported by relational database System.

This also means that NoSQL databases, which don't support those features, trade consistency for performance and scalability.

NO SQL Databases Types-Document data store

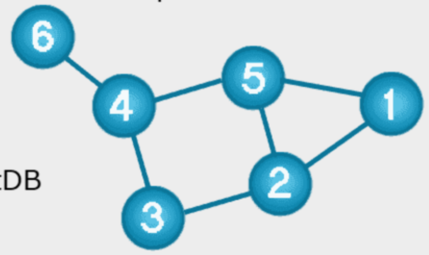
Document Store





NO SQL Databases Types-Graph data store type

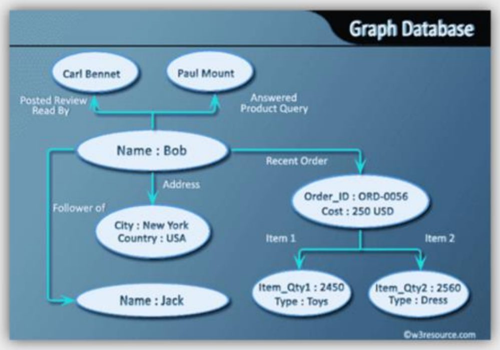
- Graph databases store all of their information in nodes (vertices) and edges.
- Each node represents an entity (such as a student or business) and each edge represents a connection or relationship between two nodes.
- Relationship information about nodes is stored in the edges.
- Graph traversal is how you "query" the database.
- Example of Graph databases : OrientDB

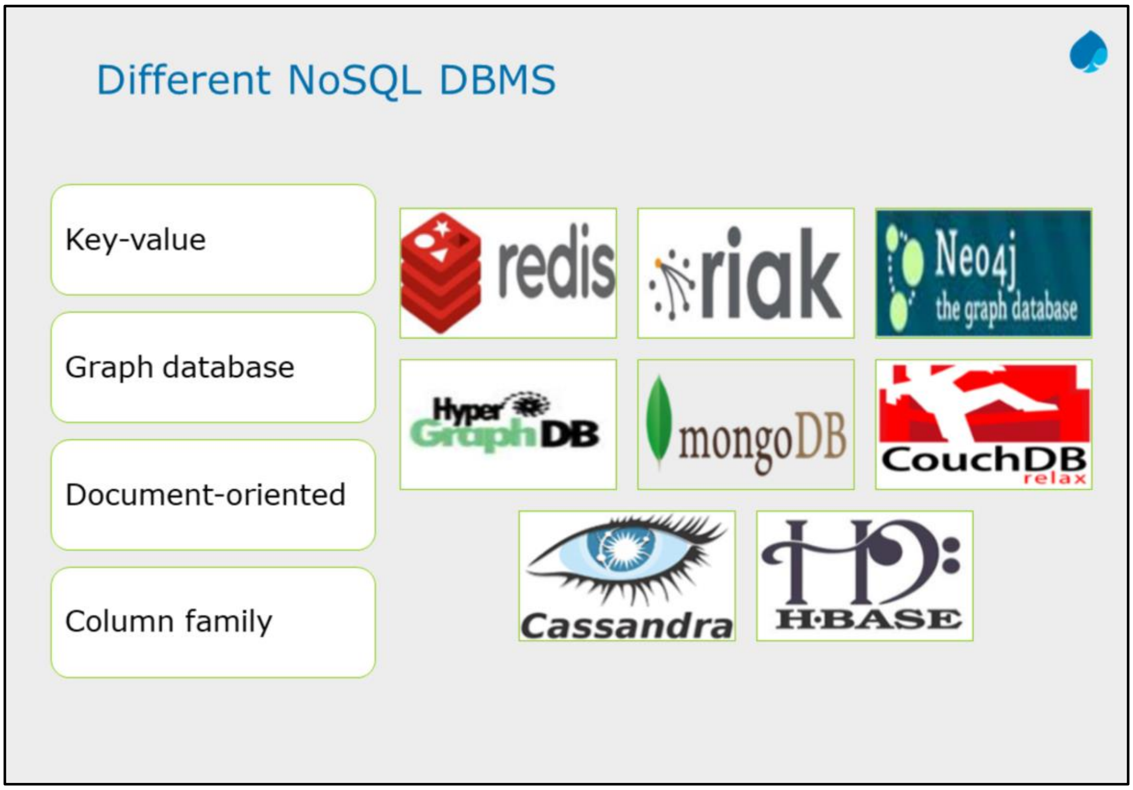


Most NoSQL databases do not support *reliability features(ACID)* which are natively supported by relational database System.
This also means that NoSQL databases, which don't support those features, trade consistency for performance and scalability.

NO SQL Databases Types-Graph data store type

Graph Database





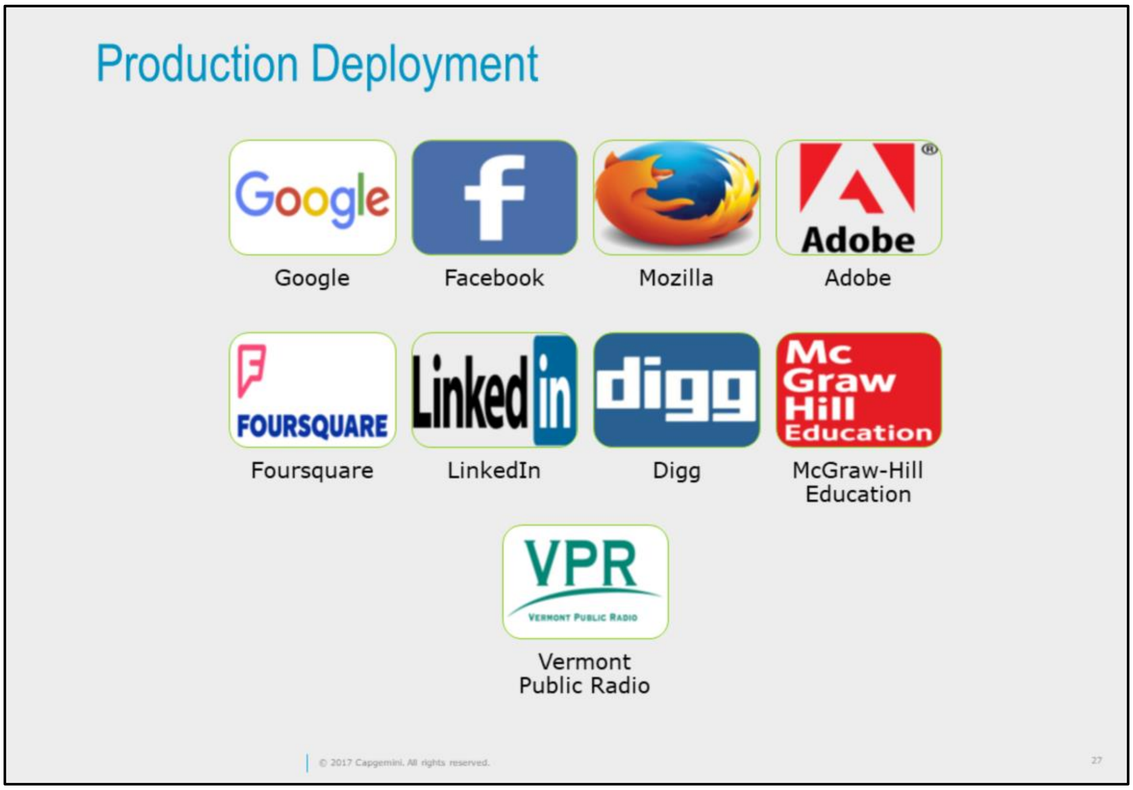
Different NoSQL DBMS

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

redis
riak
Neo4j the graph database
HyperGraphDB
mongoDB
CouchDB relax
Cassandra
HBASE

•Huge quantity of data => Distributed systems => expensive joins =>
•New fields, new demands (graphs) =>

Different data strucutres:
Simplier or more specific



Production Deployment

Google

Facebook

Mozilla

Adobe

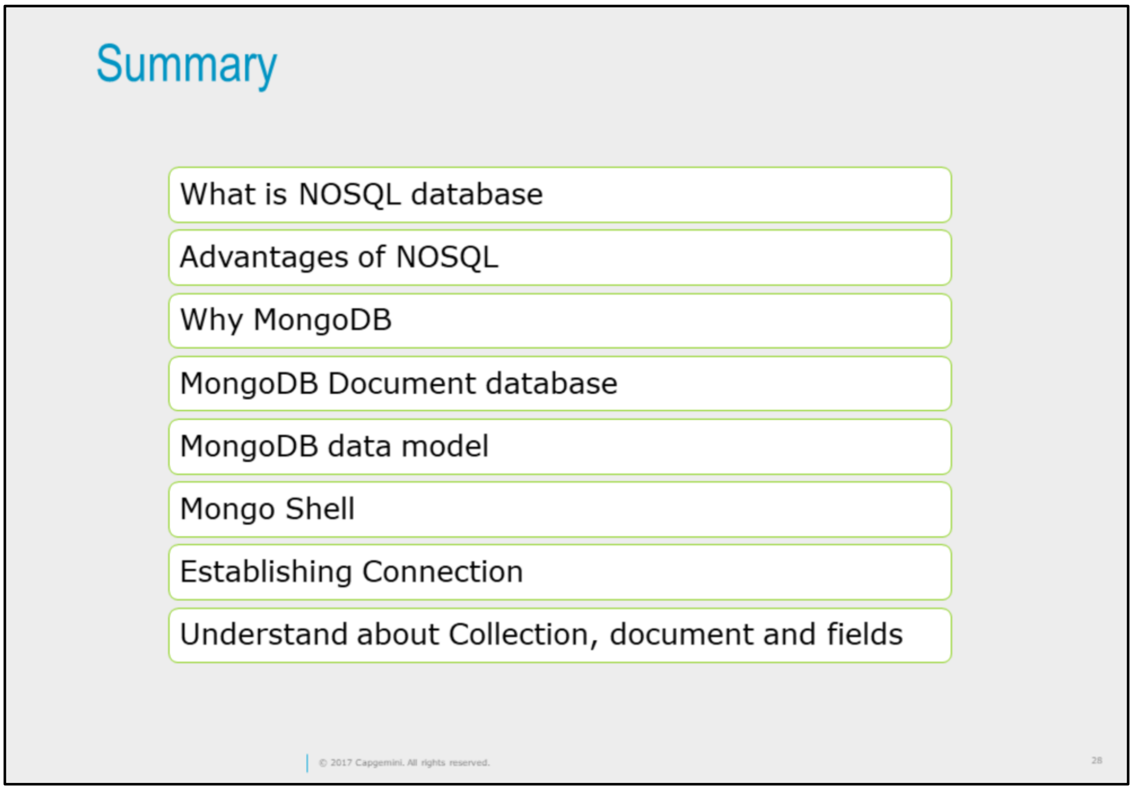
Foursquare

LinkedIn

Digg

McGraw-Hill Education

Vermont Public Radio



Summary

- What is NOSQL database
- Advantages of NOSQL
- Why MongoDB
- MongoDB Document database
- MongoDB data model
- Mongo Shell
- Establishing Connection
- Understand about Collection, document and fields