

Data Stores

Hanoi – Autumn 2021

RDBMS vs NoSQL

- Relational database is a digital database base on relational model of data
- Relational model organizes data into one or more relation (table) of tuples (row)
- Non-Relational database or NoSQL DB(Not Only SQL) provides a mechanism for storage and retrieval of data that is modeled other than tabular relational used in relational DB.



RDBMS

No	Vendor	Description
1	Oracle	Oracle RDBMS is the market leader in the DBMS market. They also owns and provides an open-source MySQL database. Oracle 12c supports columnar data stores as well as the native JSON data type.
2	Microsoft	Microsoft SQL Server is among the market leaders in the DBMS market. SQL Server 2016 supports columnar data stores as well as the native JSON data type. Microsoft Azure SQL Database is Microsoft's cloud-based offering.
3	Teradata	It is mainly suitable for building large scale data warehousing applications
4	Amazon	<ul style="list-style-type: none">- Amazon Aurora is Amazon's optimized MySQL-compatible relational database.- Amazon Relational Database Service (RDS) is a web service that allows users to run leading RDBMs such as Oracle, SQL Server, PostgreSQL, Amazon Aurora, MySQL and MariaDB. MariaDB is a fully compatible fork of MySQL. Amazon RDS provides the ability to scale databases and automate common database administration tasks.- Redshift is Amazon's relational data warehouse and is a major player in big data implementations due to its ability to store petabyte-sized databases.

RDBMS

No	Vendor	Description
5	IBM	DB2 has been IBM's main relational database and is available on servers running Microsoft Windows and Linux as well as on its mainframes. dashDB is IBM's managed cloud database, and it is based on the DB2 engine. Netezza Analytics is IBM's data warehouse appliance. It was built on PostgreSQL but does not maintain compatibility.
6	Google	Google Cloud SQL is a managed MySQL offering (PostgreSQL and SQL Server will be available in 2017).
7	PostgreSQL	One of the most popular open-source relational databases, it has support for object semantics, JSON document store and key-value data types.
8	MemSQL	MemSQL uses a combination of in-memory and disk storage to deliver very fast ingest. It provides near-real-time data ingest and processing. It is available on-premises and in the cloud. MemSQL has launched its managed cloud service.
9	VoltDB	VoltDB is a highly available and scalable in-memory RDBMS that scales across large numbers of nodes

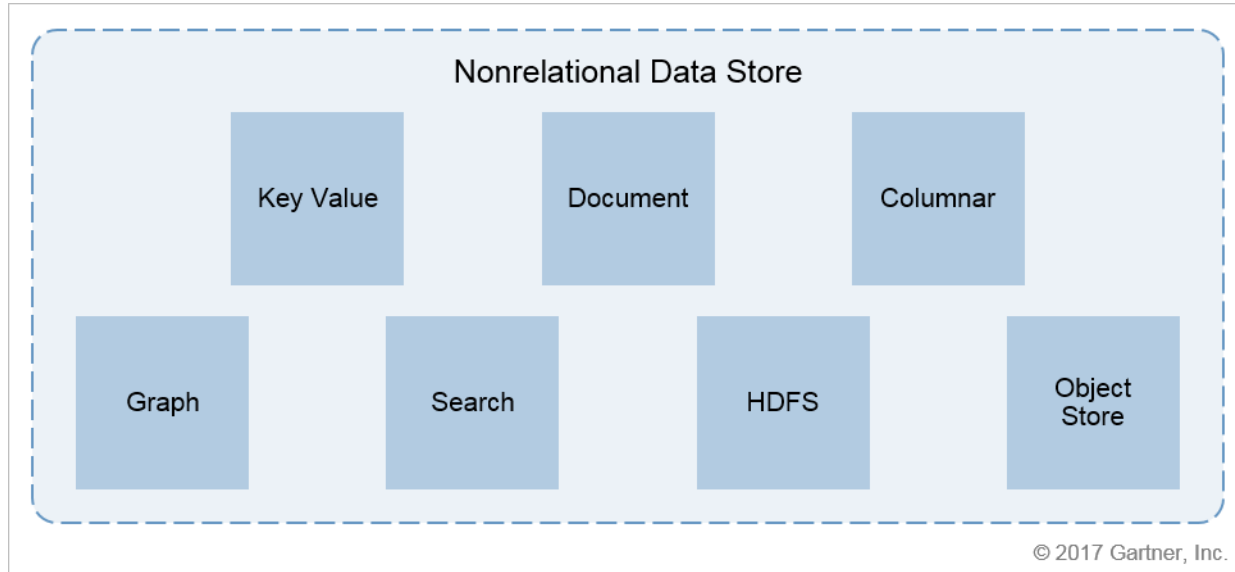
RDBMS vs NoSQL (2)

No	RDBMS	NoSQL
1	ACID: Support for transactions is part of the core design	Limitation and adhere to “eventually consistency”
2	Has rigid schema which is defined during the creation of tables. This is known as early binding or schema-on-write	Schema doesn't have to be defined in advance. This is known as late binding or schema-on-read
3	They are best suited for structured data .	Handles structured, semi-structured and unstructured data.
4	Relational online transaction processing (OLTP) systems were designed to run on a single machine. Scalability requires the machine to be "scaled up" or "scaled vertically." This necessitates adding more CPUs and memory. Scalability is limited for OLTP relational database management system (RDBMS), but massively parallel RDBMS data warehouses scale out.	Designed to be distributed. Scalability is achieved by adding more nodes to the cluster. This is also called "scaling out" or "scaling horizontally." It is preferred for applications with large concurrency or high throughput needs.

RDBMS vs NoSQL (3)

No	RDBMS	NoSQL
5	Data Access is via Structured Query Language (SQL) and Open Database Connectivity (ODBC)/Java Database Connectivity (JDBC) or native APIs.	Data access is via APIs, but SQL is becoming more prevalent.
6	Query patterns contains complex SQL and include joins and aggregations. They are better suited for ad hoc queries . Their <i>focus is on data</i> .	Not many support join operation.
7	Performance is achieved by scale-up only partitions and indexes in RDBMS OLTP databases.	Performance is achieved by scale-up and scale-out partitions, sharding and indexes

NoSQL



Key Value Store

- It simply a map of pairs of keys and values. Values can be any of well-understood datatypes such as integer, string, string, array.
- Advantages:
 - Ability to handle workloads requiring extremely high throughput and/or demanding concurrency and performance loads
 - Scalability is easy to achieve as the keys can be partitioned across nodes in a cluster
 - No knowledge of data or schema is needed beforehand.



Key Value Vendors / Products

No	Vendor/ Product	Features
1	Redis	<ul style="list-style-type: none">• Redis is one of the most widely deployed databases in the key-value category. It also supports document and time-series. As the product has matured, it has added support for complex data types in addition to string keys and string values.• It stores data in memory to provide high performance and scalability. Optional disk-based (including flash drives) persistence is available.• It is available on-premises and in the cloud (private and public) by Redis Labs in open-source and commercial versions. Can be used to complement data processed in memory by Apache Spark.
2	Riak	<ul style="list-style-type: none">• Riak is highly available and fault tolerant due to its master-less distributed architecture. Good candidate for extremely low downtime needs.• It is available on-premises and in the cloud

Key Value Vendors / Products


No	Vendor/ Product	Features
3	Memcached	<ul style="list-style-type: none">• Memcached was an early pioneer in this category. It is open-source key-value in-memory cache and is better-suited for smaller and static data compared to Redis.• It has multithreaded architecture, as opposed to single threaded Redis
4	Hazelcast	<ul style="list-style-type: none">• Hazelcast is optimized as in-memory data grid (IMDG). Works with recently announced open-source distributed data processing engine called Jet.• It is optimized to ingest high-volume data.• Hazelcast is available in open-source and commercial version in on-premises and cloud models.
5	Aerospike	Aerospike is optimized to store indexes in memory and data in memory or on solid-state drives (SSD) to provide low query latency and high scalability.

When / When Not to Use Key Value Data Store

- Low latency writes or high insert performance
- Storing high frequency inserts such as web session logs
- Not for applications that require complex queries or SQL access



Document Data Store

- Data is stored in a record that could be written in a self-describing format such as HTML, XML or JSON. Each record contains a complete set of information and has no external references
 - Each document has a primary key. It is important to model document databases based on the concurrency and read/update requirements
 - Document databases can provide ACID compliance within a single document only. Data access is through APIs. These databases provide very low read latency.
 - In order to achieve high performance and scalability, document databases are "sharded" across nodes in the cluster. Sharding is usually based on the hash key of the document or the document ID. High availability is achieved through replication. Each shard contains primary and secondary replicas.
 - Because JSON files can be verbose and large, efficiency is gained by compressing the files into binary JSON (BSON).
- 

Document Data Store Vendors/Product

No	Vendor/Product	Features
1	MongoDB	<ul style="list-style-type: none">• MongoDB is ranked as the most popular NoSQL data store.• It has excellent partitioning of data using sharding.• Its initial teething problems have been fixed as the product has matured.• It is highly popular due to ease of development and use.• A new hosted option is available from the company or from third-party providers.
2	Amazon DynamoDB	<ul style="list-style-type: none">• DynamoDB is only available as a hosted option in Amazon Web Services (AWS) and excels as a fully managed offering. It is on a very strong upward growth and is among the most popular cloud databases.• It supports not only document but also key-value data structures and graph via TitanDB plug-in.• It is very well-integrated with the rest of the AWS stack offerings.

Document Data Store Vendors/Product

No	Vendor/Product	Features
3	Apache CouchDB	<ul style="list-style-type: none">• CouchDB is highly scalable and available.• It provides an easy onramp from a single-server instance to highly clustered.• Large installations use Apache CouchDB for performance.
4	Couchbase	<ul style="list-style-type: none">• Couchbase is available as open-source and distributed (earlier known as Membase) options and has been developed by the same team as Apache CouchDB.• It is compatible with the popular key-value store called memcached, thus providing both low-latency reads via in-memory caching and disk storage for persistent storage.• Its commercial version is deployed at large enterprise clients to handle big data needs.

Document Data Store Vendors/Product

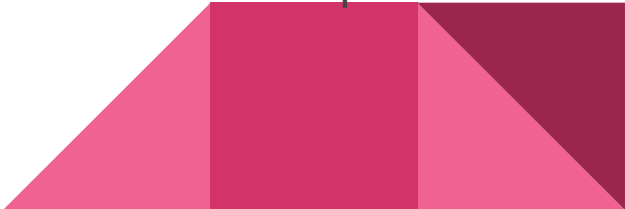
No	Vendor/Product	Features
5	IBM Cloudant	Cloudant shares the same core as Apache CouchDB. Its hosted version is available from IBM.
6	MarkLogic	MarkLogic started as an XML database, but it provides support for document and search data stores while providing full ACID compliance.
7	Microsoft Azure DocumentDB	Microsoft's Azure-based fully managed offering has seen a strong growth. It is compatible with MongoDB, thus providing applications written in MongoDB a migration path to Azure cloud.
8	Google Cloud Datastore	Google Cloud Datastore is fully managed with automatic support for sharding and replication. It is highly scalable and consistent.

When / When Not Use Document Data Store

- Low-latency reads and easy migration from RDBMS
- Best for applications with dynamic data that need a schema-less data store
- Not for applications that require strong consistency



Column-Oriented Data Store

- It organizes similar columns in a manner that provides very high throughput.
 - Columns can be added on-the-fly, and the schema need not be defined in advance.
 - Data is still stored in rows, and the primary key is still the row key, which is used to partition data across nodes within a cluster. However, instead of storing the same number of columns in the row, as in a row store relational database, each row may have a different number of columns, which reduces the problem of sparse tables
- 

Vendor/Product

No	Product/Vendor	Feature
1	Apache Cassandra	<ul style="list-style-type: none">• Cassandra is available as open source from Apache or via a supported version from DataStax.• It was originally developed by Facebook and based on Amazon DynamoDB and Google Bigtable.• It has high scalability and performance.• Data access is via SQL-like Cassandra Query Language (CQL).• In addition to columns, it supports key-value, tabular and now graph (TitanDB).
2	Apache HBase	<ul style="list-style-type: none">• HBase is column-oriented key value store and an open-source implementation of Google Bigtable.• It is well-integrated with the Hadoop ecosystem. For example, HBase can be managed along with the rest of the Hadoop ecosystem components using Apache ZooKeeper (Cassandra uses its own protocol called Gossip).• Data access is via Java, Thrift and RESTful APIs.• Its replication is master-slave.
3	Apache Accumulo	a column-oriented data store with very low-grained security at cell level.

When / When Not Use Columnar Data Store

- You need a database to handle very large data volumes with scalable and highly available architecture
- New hybrid data stores where row store handles fast ingest and column store handles fast reads.
- Analyzing log data or need scanning huge, two-dimension join-less tables is a requirement



Graph Data Store

- Hierarchical data is represented in relational databases with the help of foreign keys. When this "graph of relations" grows, relational databases are no longer the most efficient way to query. Use of "joins" across multiple tables in a query can be very expensive in terms of resources and time, especially for traversing large graphs.
- Graph databases are designed to store complex hierarchical relations so they can be queried very efficiently



Graph Data Store Vendor / Product

No	Vendor/Product	Features
1	Neo4j	<ul style="list-style-type: none">• Neo4j is among the most popular graph databases. It was used to find interconnections in the "Panama Papers" scandal.• Open-source and commercial versions are available from Neo.
2	OrientDB	<ul style="list-style-type: none">• OrientDB is a hybrid of document and graph database.• It is open source.
3	TitanDB	<ul style="list-style-type: none">• TitanDB has been bought by DataStax, and the technology has been incorporated inside DataStax Enterprise (DSE).• The team behind TitanDB has launched a fork of TitanDB called JanusGraph, which is in early stages of development.

When / When Not Use Graph Data Store

- To store relationship-rich data with very high performance on large amounts of data.
- Not to be used by applications that require SQL JOIN type of operations.



Search Data Store

Search data stores provide advanced search capabilities such as wildcard searches, grouping and ranking searches, and drill-down capabilities. They deploy inverted indexes whereby each unique term in the indexed documents is stored in a "dictionary," which has a pointer to the unique document ID where the terms exist and are called "postings." Search databases are designed for a specific purpose and hence shouldn't be used for use cases that require ACID compliance



Search Data Store Vendor / Product


No	Vendor/Product	Features
1	ElasticSearch	<p>Elasticsearch is open source and is based on Apache Lucene Core, but it is managed by Elastic. Elastic also provides a commercial version with enhanced features such as security, monitoring, visualization and graph analytics.</p> <p>It is the most widely deployed distributed search database.</p> <p>Its parent company, Elastic, bundles its other products, Logstash and query tool Kibana in a stack known as Elastic (previous called ELK).</p>
2	Apache Solr	<p>Solr is also open source and is based on Apache Lucene with wide community support.</p> <p>It is best suited for full-text searching.</p>
3	Splunk	<p>Splunk is a proprietary search engine that predates both Solr (2004) and Elastic (2010).</p>

Object Store

- Object stores are used to store raw data on a storage device, thereby reducing the overhead of a file system or a data store.
- The files could be raw logs being generated by web servers, comma-separated value (CSV) files, videos, pictures, backup data, IOT sensor data, clickstreams and so on



Vendor/Product

 Product/Vendor	Description
Amazon S3 (Simple Storage Service)	<ul style="list-style-type: none">Amazon S3 is the backbone of Amazon's data infrastructure and is the most widely used object store.Its durability is 11 9's, and its availability is 4 9's. Amazon S3 provides synchronous replication of data across multiple facilities and automatic data integrity checks.Amazon Athena allows data in Amazon S3 to be directly queried.It is integrated with Amazon's lowest-cost storage called Glacier for "cold data."
Microsoft Azure Blob Storage	<ul style="list-style-type: none">Azure Blob Storage (previously Windows Azure Storage Blob [WASB]) is a general-purpose object store used by applications such as the Azure Data lake Store (ADLS).Data replication is asynchronous.Data access is via REST APIs and client libraries for all major programming languages.
Google Cloud Storage	<ul style="list-style-type: none">Large data is stored in Google Cloud Storage and accessed via a RESTful web service. Data is read-only.Google storage stores data in JSON or CSV format. Queries are based on SQL, but data is returned in JSON.Google BigQuery allows interactive analysis of read-only data. The underlying technology for the ad hoc query engine is called Dremel.

When / When Not to Use Object Store

- Low cost, especially for large binary content such as videos.
- Ideal as cloud-based data stores.
- Common SQL-on-Hadoop engines such as Apache Drill, Hive, Amazon Athena, Impala and Presto work with object stores.



Appendix: CouchDB vs Couchbase

	Couchbase Server	Apache CouchDB
Data models	Document, Key-Value	Document
Storage	Append-only B-Tree	Append-only B-Tree
Consistency	Strong	Eventual
Topology	Distributed	Replicated
Replication	Master-Master	Master-Master
Automatic failover	Yes	No
Integrated cache	Yes	No
Memcached compatible	Yes	No
Locking	Optimistic & Pessimistic	Optimistic with MVCC
MapReduce (Views)	Yes	Yes
Query language	Yes, N1QL (SQL for JSON)	No
Secondary indexes	Yes	Yes
Notifications	Yes, Database Change Protocol	Yes, Changes Feeds

Appendix: Columnar Storage

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

```
+---+-----+-----+-----+-----+-----+-----+
| ID | name      | address          | zip code | phone   | city      | country | age |
+---+-----+-----+-----+-----+-----+-----+
| 1 | Benny Smith | 23 Workhaven Lane | 52683    | 14033335568 | Lethbridge | Canada  | 43 |
| 2 | Keith Page  | 1411 Lillydale Drive | 18529    | 16172235589 | Woodridge  | Australia | 26 |
| 3 | John Doe    | 1936 Paper Blvd.   | 92512    | 14082384788 | Santa Clara | USA     | 33 |
+---+-----+-----+-----+-----+-----+-----+
```

```
1,Benny Smith,23 Workhaven Lane,52683,14033335568,Lethbridge,Canada,43;2,Keith Page,1411
```

A1	A2	A3	B1	B2	B3	C1	C2	C3
----	----	----	----	----	----	----	----	----

```
1,2,3;Benny Smith,Keith Page,John Doe;23 Workhaven Lane,1411 Lillydale Drive,1936 Paper Blvd.;
```

Appendix: Columnar Storage Ad/dis

Advantages

- Since the data type for each column is similar, so get better compression when running compression algorithms on each column.
- I/O will be reduced as we can efficiently scan only a subset of the columns (for example: query the average age of all users)

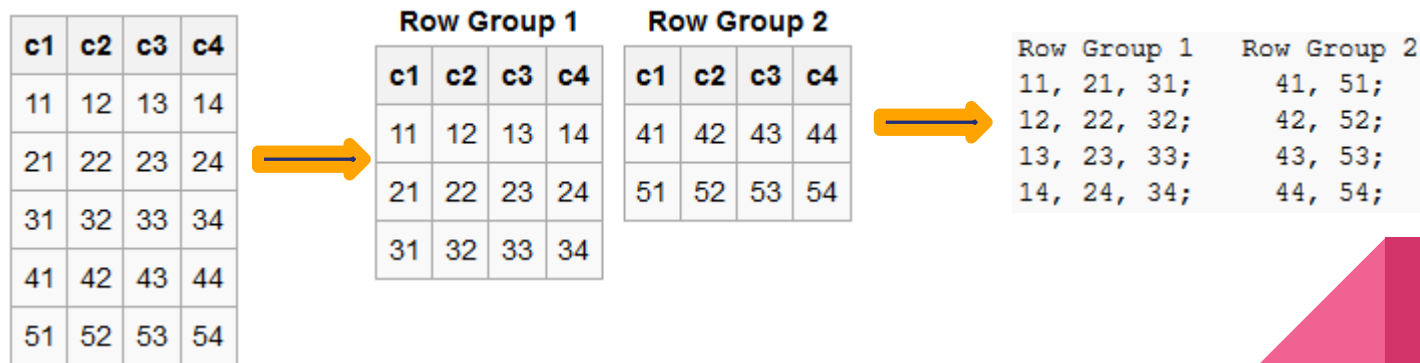
Disadvantages

- Incremental data loading
- Queries against only a few row



Appendix: RCFile

- RCFile (Record Columnar File) is a data placement structure designed for MapReduce-based data warehouse systems
- RCFile stores table data in a flat file consisting of binary key/value pairs. It first partitions rows horizontally into row splits, and then it vertically partitions each row split in a columnar way



Appendix: Apache Parquet

- Parquet is a columnar storage format available to any project in the Hadoop ecosystem.
- Originally developed by a collaboration between Twitter and Cloudera and based on Google Dremel.

Design goals

- Reduce storage cost for large datasets with stable schemas
 - Reduce required IO for queries
 - Flexible compression options. The data can be compressed with any of several codecs. The compression is transparent to applications that read the data files
 - Large file size. The layout of Parquet data files is optimized for queries that process large volumes of data, with individual files in the multi-megabyte or even gigabyte range
- 