

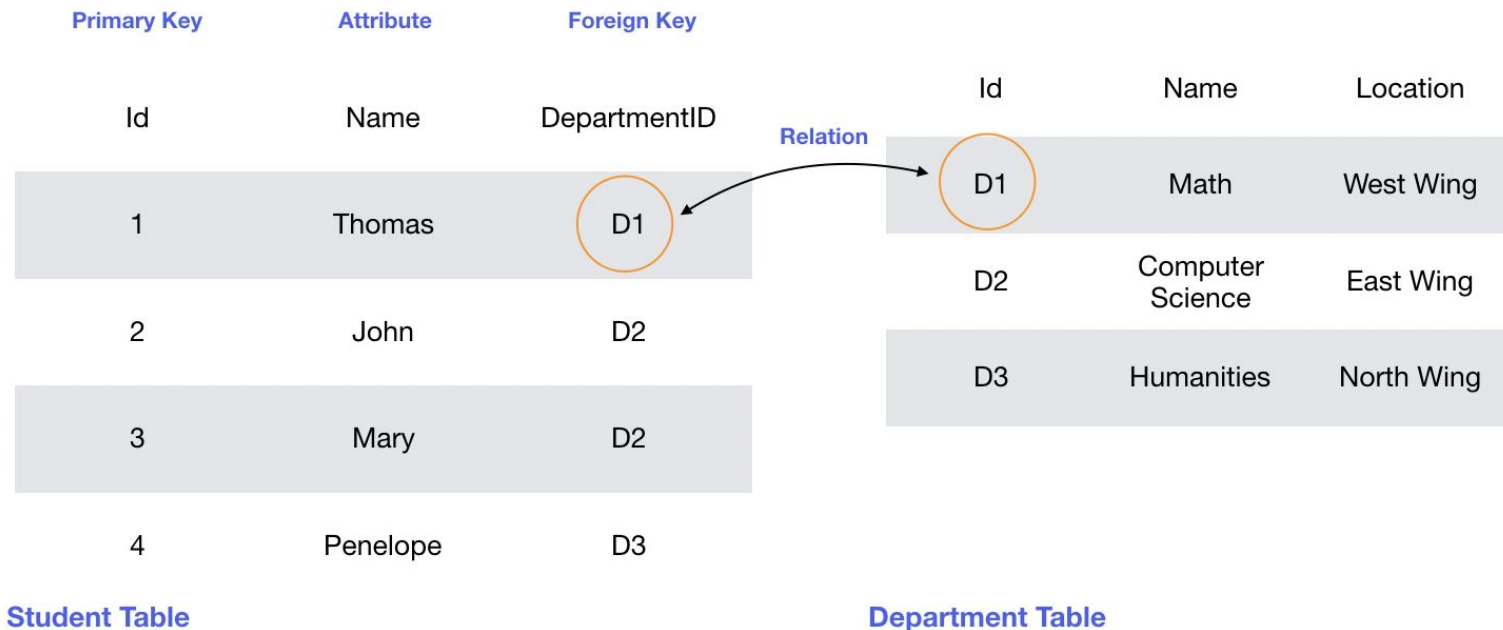
SQL vs NoSQL in terms of scalability

- Kishor Mishra (2019201038)
- Pushkar Talwalkar (2018101010)
- Himanshu Maheshwari (20171033)

What is SQL?

1. SQL stands for Structured Query Language. When people refer to SQL Databases, they're actually referring to Relational Database Management System (RDBMS).
2. Data is stored in the form of tables. Each row is uniquely identified by a primary key and each row can also have a foreign key, which is the primary key of another table. Through the foreign key, a relation is formed with another table — hence the notion of relational databases.
3. Examples of SQL databases include MySQL, PostgreSQL and Oracle Database.

What is SQL?

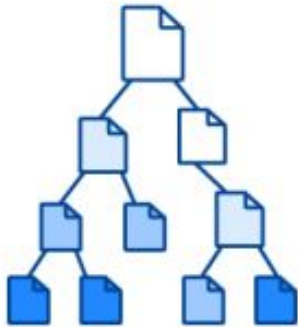


What is NoSQL?

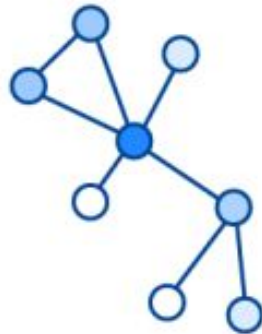
1. NoSQL is known as “No SQL,” or sometimes “Not Only SQL.”
2. NoSQL databases support different ways of storing the data as opposed to storing them in a table in SQL databases.
3. Broadly there are four types of NoSQL databases:
 - Document (e.g. MongoDB, Firestore).
 - Graph (e.g. Neo4j).
 - Key-value (e.g. Riak, Berkeley DB).
 - Wide-column (e.g. Cassandra, HBase).

What is NoSQL?

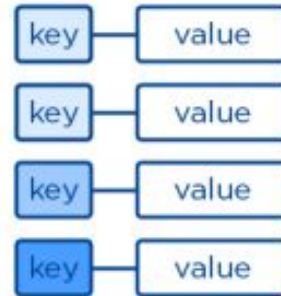
Document



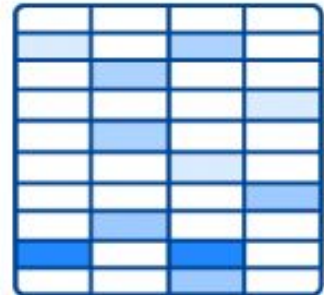
Graph



Key-Value



Wide-column



Difference between SQL and NoSQL?

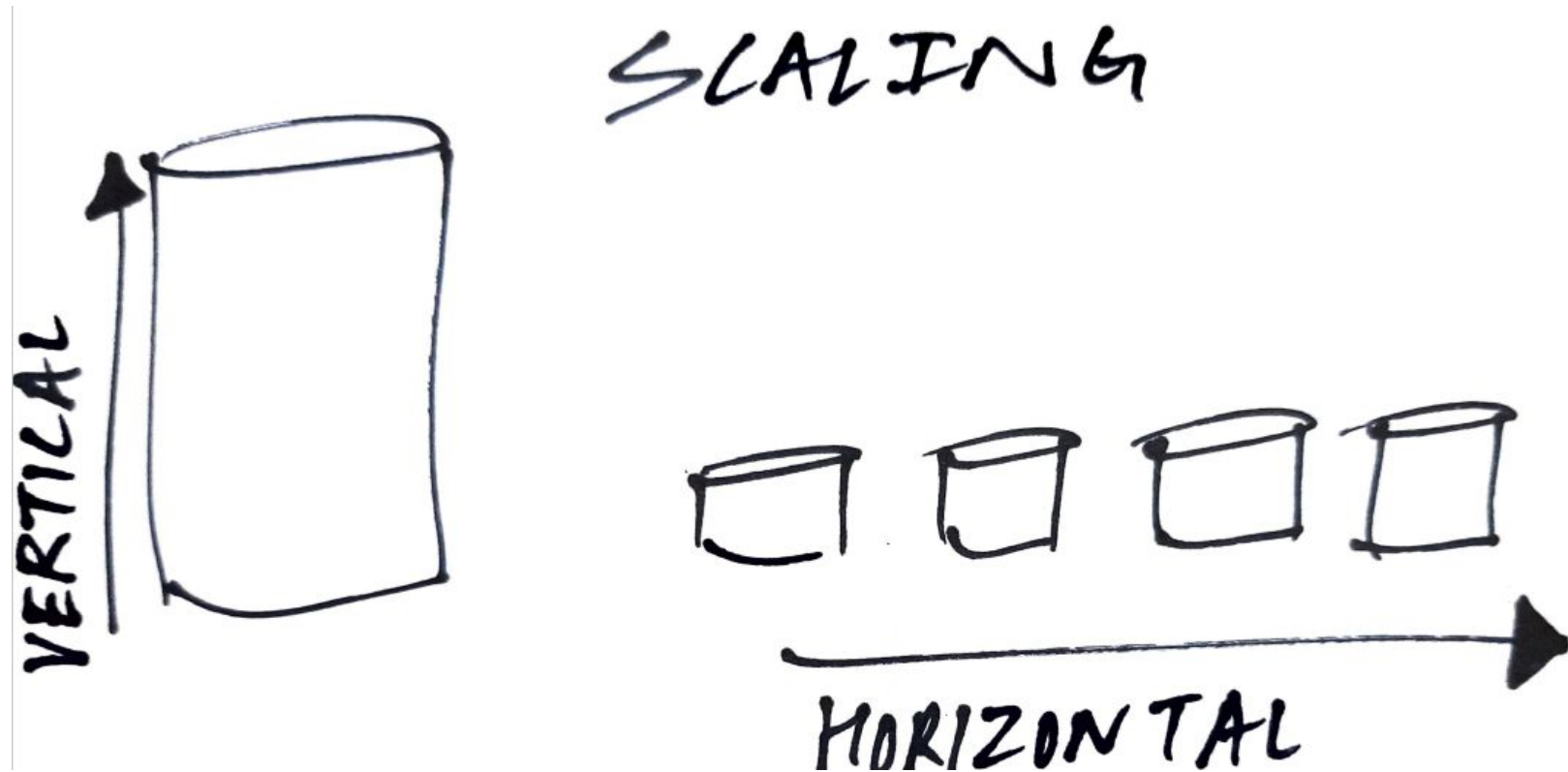
SQL	NoSQL
SQL databases have fixed or static or predefined schema.	NoSQL databases have dynamic schema.
SQL databases display data in form of tables so it is known as table-based database.	NoSQL databases display data as collection of key-value pair, documents, graph databases or wide-column stores.
SQL databases are vertically scalable.	NoSQL databases are horizontally scalable.
SQL databases use a powerful language "Structured Query Language" to define and manipulate the data.	In NoSQL databases, collection of documents are used to query the data. It is also called unstructured query language. It varies from database to database.
SQL databases are best suited for complex queries.	NoSQL databases are not so good for complex queries because these are not as powerful as SQL queries.

Scalability

Vertical scaling means if you exceed the current capacity of your server, you would have to use a more powerful CPU, add more RAM, stack up the storage, etc.

Horizontal scaling means if you run out of capacity, you can simply add a machine to the cluster (a bunch of machines working together). These machines are usually much cheaper.

Scalability



SQL vs No SQL in terms of Scalability

1. SQL databases supports vertical scaling.
 2. They run on traditional machines i.e. a single server and can only support vertical scaling.
-
1. NoSQL databases usually run on a number of interconnected machines which is known as a cluster.
 2. The data is distributed amongst the machines within the cluster. Each machine will store some portion of the data.
 3. Thus they support horizontal scaling.

SQL vs No SQL in terms of Scalability

1. Since NoSQL is just storage of self contained objects, there is no need for JOIN requests, and thus sharding of the database is easy.
2. On the other hand, RDBMS requires JOIN requests etc so vertical stacking is the only option.

SQL supports ACID property

1. **Atomicity** – Transactions are performed one at a time or they don't happen at all.
2. **Consistency** – This ensures that the database is not left in half a complete state. If an error occurs, then it ensures that the roll back changes takes place.
3. **Isolation** – Transactions occur independently. No transaction has access to any other transaction.
4. **Durability** – The changes made to the database through transactions on completion are committed to the database and the updates are not lost.

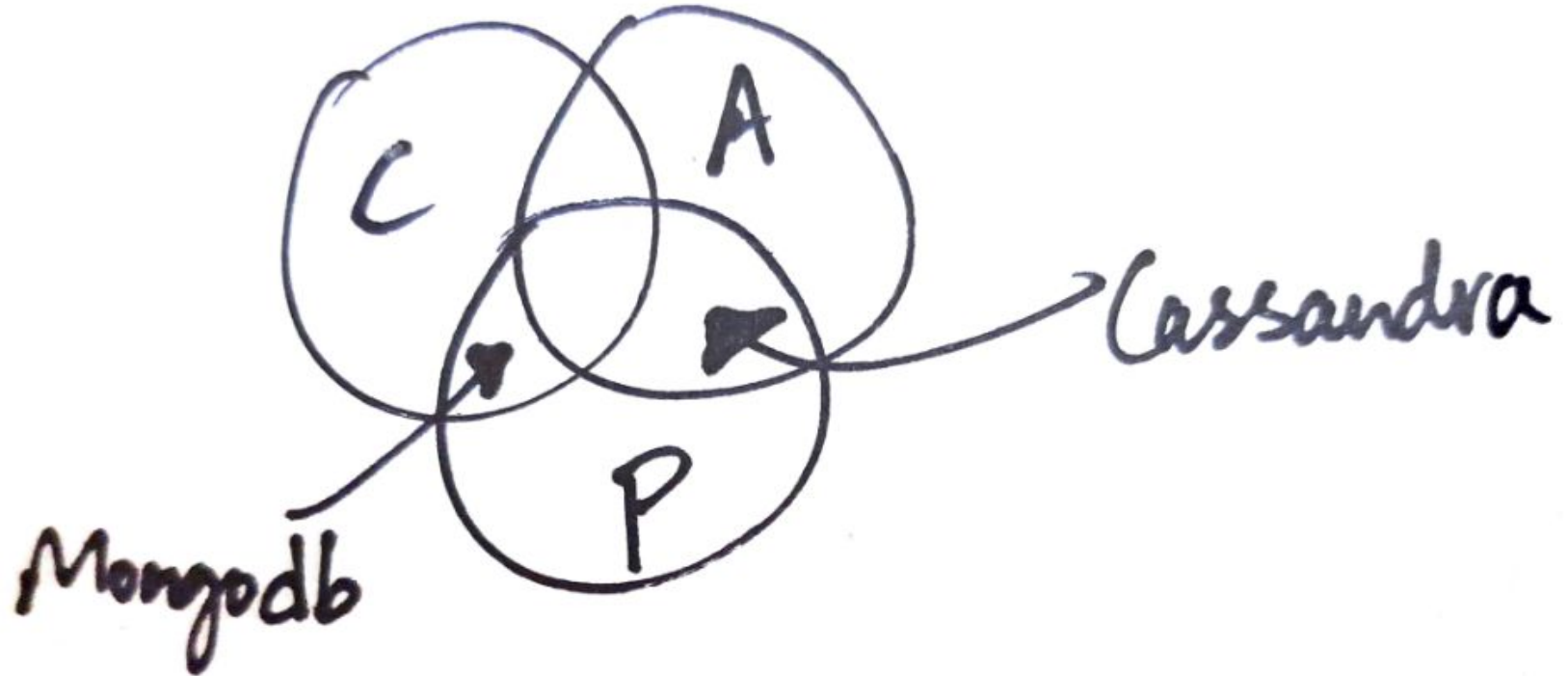
NoSQL supports BASE property

1. NoSQL databases are designed with scale in mind and therefore adhere to BASE properties instead of ACID.
2. **Basic Availability** - The database favors the availability of data rather than the correctness of the data.
3. **Soft-state** - The database stores (e.g. replicas) does not have to be consistent all the time.
4. **Eventual Consistency** - After a period of time, the database stores will eventually be mutually consistent.

NoSQL and CAP theorem

1. NoSQL do not provide ACID properties entirely.
2. Since it supports horizontal scaling, the CAP theorem governs them.
3. CAP theorem says - A distributed database system can only have 2 of the 3:
 - a. Consistency
 - b. Availability
 - c. Partition Tolerance.

NoSQL and CAP theorem



Conclusion

1. **By virtue of its design NoSQL is more preferred databases for distributed system.**
2. **Also, because of horizontal scaling, NoSQL is better at scaling than SQL. Scaling SQL horizontally is possible but is difficult due to ACID property of SQL.**
3. **It does so by not following the rigid ACID rules, and offers eventual consistency instead.**

Thank -you

References:

1. <https://medium.com/better-programming/introduction-to-nosql-databases-7f6ed6e055c5>
2. <https://medium.com/better-programming/scaling-sql-nosql-databases-1121b24506df>
3. <https://blog.panoply.io/sql-or-nosql-that-is-the-question>
4. <https://stackoverflow.com/questions/8729779/why-nosql-is-better-at-scaling-out-than-rdbms>
5. <https://www.analyticsvidhya.com/blog/2020/10/sql-vs-nosql-databases-a-key-concept-every-data-engineer-should-know/>