SQL & NoSQL Difference

|  |  |
| --- | --- |
| sql | no-sql |
| relational database | non-relational database |
| pre-defined schema | dynamic schema |
| vertical scaling | horizontal scaling |
| ACID | CAP |
| not suited for hierarchical data store | suited for hierarchical data store |

Relational database: A relational database is **a collection of data items with pre-defined relationships between them**. These items are organized as a set of tables with columns and rows. Tables are used to hold information about the objects to be represented in the database.

Non-relational database: A non-relational database is **a database that does not use the tabular schema of rows and columns found in most traditional database systems**. Instead, non-relational databases use a storage model that is optimized for the specific requirements of the type of data being stored.

Examples for each kind of No-sql

key/value: redis, riak

graph: Neo4j, GraphDB

document: mongoDB, CouchDB

columnar : cassandra, Hbase

pre-defined schema: pre-defined schema refers to that structure never changing. A dynamic schema is one that changes as you add data.

The **scalability** of an application can be measured by the number of requests it can effectively support simultaneously. The point at which an application can no longer handle additional requests effectively is the limit of its scalability.

Horizontal scaling means scaling by adding more machines to your pool of resources (also described as “scaling out”), whereas vertical scaling refers to scaling by adding more power (e.g. CPU, RAM) to an existing machine (also described as “scaling up”).

Diagram

Description automatically generated

ACIDA picture containing diagram

Description automatically generated

Atomicity:

—**Abort**: If a transaction aborts, changes made to database are not visible.   
—**Commit**: If a transaction commits, changes made are visible.   
Atomicity is also known as the ‘All or nothing rule’.

Consistency:

Total **before T** occurs = **500 + 200 = 700**.   
Total **after T occurs** = **400 + 300 = 700**.

Isolation:

This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of database state. Transactions occur independently without interference. Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order.

Durability:

This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs. These updates now become permanent and are stored in non-volatile memory. The effects of the transaction, thus, are never lost.

CAP:

Diagram

Description automatically generated

A hierarchical database is a data model in which data is stored in the form of records and organized into a tree-like structure, or parent-child structure, in which one parent node can have many child nodes connected through links.