|  |  |
| --- | --- |
| 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.

pre-defined schema vs dynamic schema

Static schema refers to that structure never changing. A dynamic schema is one that changes as you add data. In most cases, the textbook (correct) approach uses a static schema. Some cases work out well with a dynamic schema.

vertical scaling vs horizontal scaling

While horizontal scaling refers to adding additional nodes, vertical scaling describes adding more power to your current machines. For instance, if your server requires more processing power, vertical scaling would mean upgrading the CPUs. You can also vertically scale the memory, storage, or network speed.

ACID

ACID is an acronym that refers to the set of 4 key properties that define a transaction: Atomicity, Consistency, Reliability, and Durability. If a database operation has these ACID properties, it can be called an ACID transaction, and data storage systems that apply these operations are called transactional systems.

Atomicity

Atomicity. A transaction must be Atomic, meaning all changes made by the transaction are completed as a single unit, or none of the changes are made. If a partial transaction were committed, the atomic property is violated, and the database is left in an inconsistent state.

Consistency

Consistency (or Correctness) in database systems refers to the requirement that any given database transaction must change affected data only in allowed ways. Any data written to the database must be valid according to all defined rules, including constraints, cascades, triggers, and any combination thereof.

Reliability

Database reliability is defined broadly to mean that the database performs consistently without causing problems. More specifically, it means that there is accuracy and consistency of data.

Durability

Durability in databases is the property that ensures transactions are saved permanently and do not accidentally disappear or get erased, even during a database crash. This is usually achieved by saving all transactions to a non-volatile storage medium.

CAP

Consistency

Consistency means that all clients see the same data at the same time, no matter which node they connect to. For this to happen, whenever data is written to one node, it must be instantly forwarded or replicated to all the other nodes in the system before the write is deemed ‘successful.’

Availability

Availability means that that any client making a request for data gets a response, even if one or more nodes are down. Another way to state this—all working nodes in the distributed system return a valid response for any request, without exception.

Partition tolerance

A partition is a communications break within a distributed system—a lost or temporarily delayed connection between two nodes. Partition tolerance means that the cluster must continue to work despite any number of communication breakdowns between nodes in the system.

hierarchical database

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.