

## SQL vs NoSQL

- SQL applies to relational databases. Relational database is a collection of data with predefined relationships within, the data items are organized as a table with rows and columns. Non Relational databases however, can have many different structures and handle unstructured data; they can be graph-based, document-based, key-value pairs, or wide-column stores.
- Predefined schema means that the structure of data is static, while dynamic schema can change the structure while adding items into the NoSQL database.
- Most SQL databases can be scaled vertically, by increasing the processing power of existing hardware. NoSQL databases use a master-slave architecture which scales better horizontally, with additional servers or nodes. These are useful generalizations, but it's important to note:
- SQL databases can be scaled horizontally as well, though sharding or partitioning logic is often the user's onus and not well supported.
- NoSQL technologies are diverse and while many rely on the master-slave architecture, options for scaling vertically also exist.
- Savings made using more efficient data structures can overwhelm differences in scalability; most important is to understand the use case and plan accordingly.
- ACID vs CAP
  - ACID addresses an individual node's data consistency
  - CAP addresses cluster-wide data consistency
  - ACID means Atomicity, Consistency, Isolation and Durability
    - Atomicity: guarantee that either all of the transaction succeeds or none of it does.
    - Consistency: guarantee that all data will be consistent. All data will be valid according to all defined rules, including any constraints, cascades, and triggers that have been applied on the database.
    - Isolation: all transactions will occur in isolation. No transaction will be affected by any other transaction.
    - Durability: once a transaction is committed, it will remain in the system – even if there's a system crash immediately following the transaction.
  - CAP means Consistency, Availability, and Partition-Tolerance
    - Consistency: All Nodes Have Same Data via Eventual Consistency
    - Availability: system continues to operate despite arbitrary message loss or failure of part of the system
    - Partition-Tolerance: System continues to operate despite arbitrary message loss or failure of part of the system
- The difference between relational and hierarchical databases lies in the data structures. While the hierarchical database architecture is tree-like, data in a relational database is stored in tables with a unique identifier for each record. A relational database structure facilitates easy identification and access of data in relation to other data points in the database. The tables are separate from physical storage structures, which enables

database administrators to alter physical data storage without reorganizing the database tables themselves.

- Hierarchical data store advantages and disadvantages:
  - The key advantage of a hierarchical database is its ease of use. The one-to-many organization of data makes traversing the database simple and fast, which is ideal for use cases such as website drop-down menus or computer folders in systems like Microsoft Windows OS.
  - The major disadvantage of hierarchical databases is their inflexible nature. The one-to-many structure is not ideal for complex structures as it cannot describe relationships in which each child node has multiple parents nodes. Also the tree-like organization of data requires top-to-bottom sequential searching, which is time consuming