Deciding between SQL and No SQL databases can be a difficult task for which it is important to understand the key differences between their transaction models. In this article, we are going to discuss the two most popular database transaction models: ACID and BASE.

ACID(Atomicity, Consistency, Isolation, Durability)

ACID properties are used for maintaining data integrity during transaction processing. The term ACID stands for Atomicity, Consistency, Isolation and Durability. In order to maintain consistency before and after a transaction relational databases follow ACID properties. Let us understand the different terms in the word:

Atomicity:

The **all or none principle**, either the transaction takes place or it doesn’t. If one element of the transaction fails the entire transaction fails, they don’t occur partially.

Consistency:

The transactions should follow all rules defined in the system. The database must be in a consistent state before and after a transaction.

Isolation:

Each transaction is independent of the other. A transaction does not have access to other transactions which had not finished yet. The changes occurring during a transaction will not be visible to others until it is written in the main memory.

Durability:

Once the transaction has completed and the writes and updates have been written to the disk, it will remain in the system even if a system failure occurs.

BASE(Basically Available, Soft State, Eventual Consistency)

With the rise of No SQL databases, the BASE model was designed. No SQL databases follow the CAP theorem.

Basically Available:

Basically available means that there will be a response to every request even if the response is a system failure or does not successfully return the data.

Soft State:

The state of the system is always soft which means that it can change over time even if there are no reads or writes as the system keeps changing the data to make it consistent.

Eventual Consistency:

The data might not be consistent immediately but eventually, it becomes consistent. Reads in the system are still possible even though they may not give the correct response due to inconsistency.

ACID vs BASE: Which one should you choose?

Both the models have been designed to satisfy different requirements. While choosing a database you need to decide keeping in mind the properties of both the models and the requirements of your application. **SQL** databases like MySQL, SQLite, Microsoft SQL, Oracle follow ACID properties whereas **No SQL databases**MongoDB, Cassandra, Redis follow BASE.

Apart from ACID and BASE, there are a number of other factors on which the choice of your database depends. Here is a detailed article on the same: