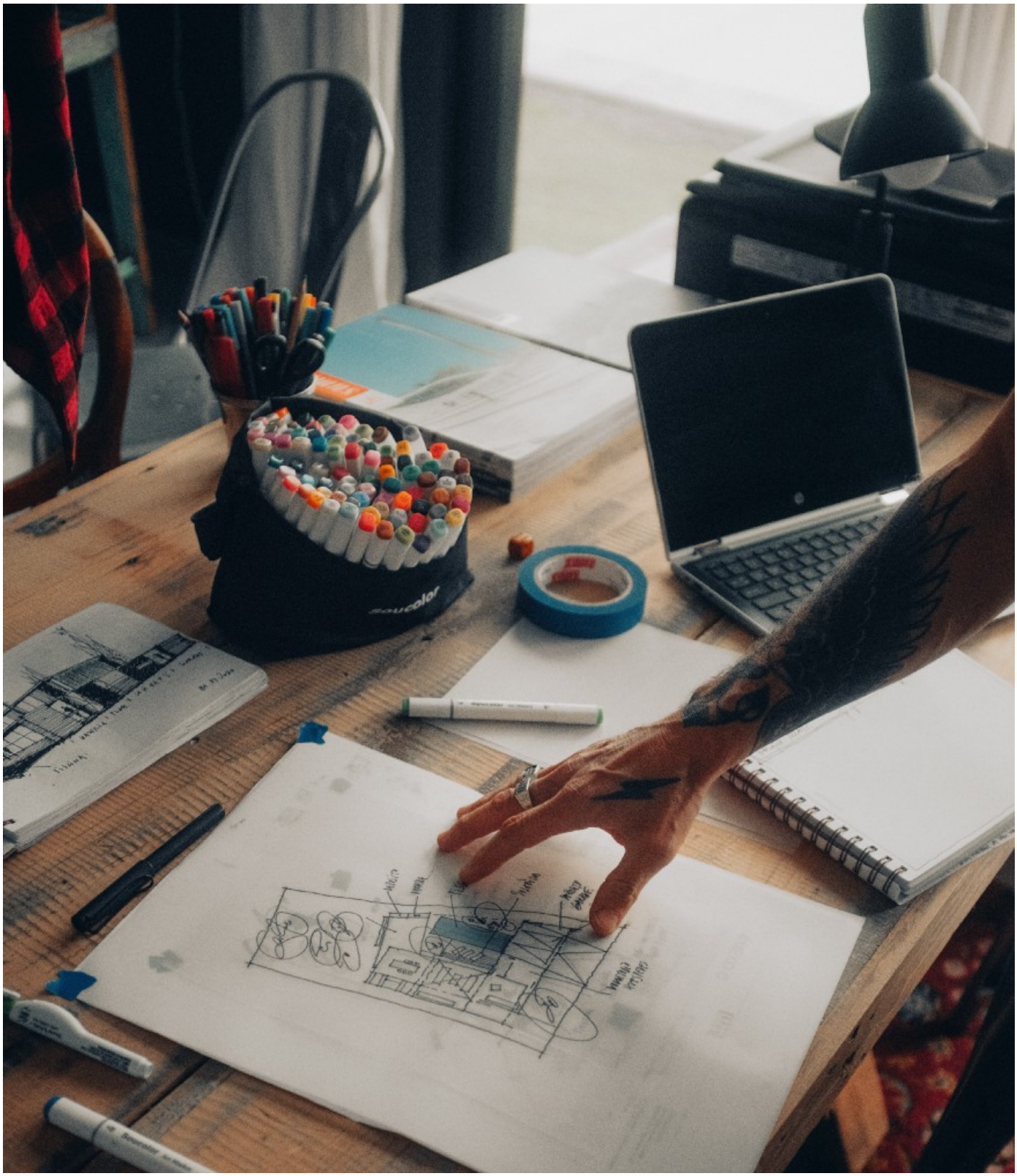
The fundamental knowledge of System Design — (2) — Database

Database System



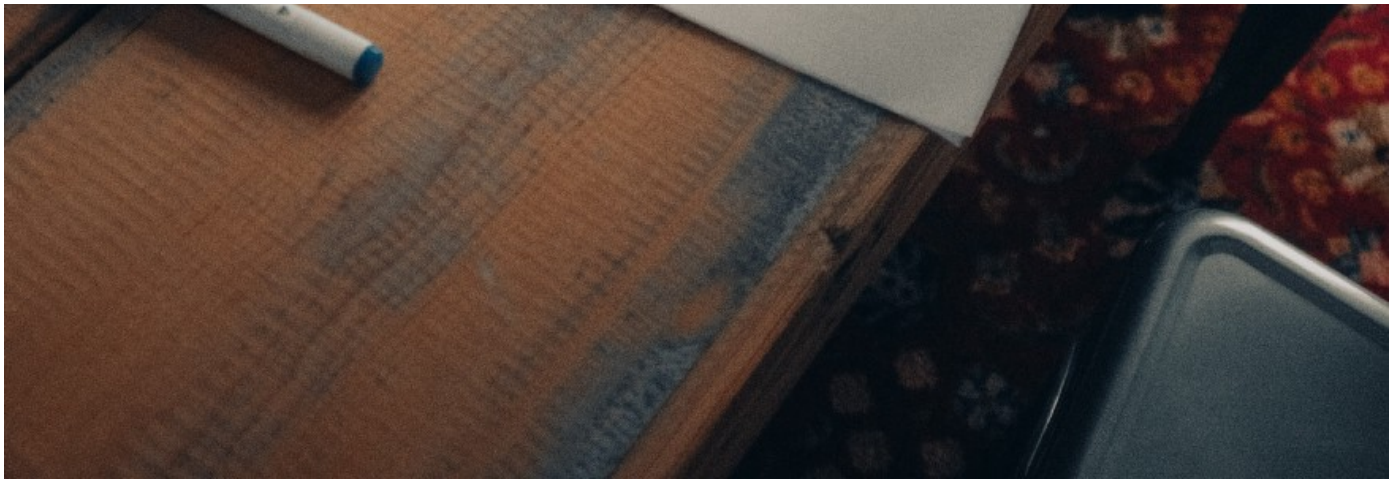


Photo by Ryan Ancill on Unsplash

The fundamental knowledge of System Design — (1)

Today, I will share the fundamental knowledge of system design.

medium.com

Database System

- The transaction is the smallest logical unit that represents one or a series of operations that occurred in the database system.
- All operations in this logical unit either succeed or fail and there is no intermediate state.
- The purpose of the transaction mechanism is no matter if the operation is successful, failed, abnormal, down, the transaction mechanism can guarantee the final consistency of the data. Even the failure of any operation will not affect the next operation. Regardless of whether the transaction is successful or a failure, the system data is ultimately consistent.

Transaction Characteristics (ACID)

1. Atomicity: Either the entire execution when the transaction is successful or the entire rollback when the transaction is a failure
2. Isolation: The execution of two transactions is independently isolated. When multiple transactions operate on an object, the second transaction can only be performed after the first transaction is completed. So, the concurrency will be minimized.
3. Consistency: The transaction must ensure the integrity and consistency of the overall database data.
4. Durability: Persistency means that once the transaction is successfully submitted, the modified data will be irreversible and persistent, it will not cause data errors or data loss due to abnormalities or downtime.

Understanding Consistency

Internal consistency (Database system): Consistency in ACID — modification loss, dirty read, non-repeatable read

External Consistency (distributed System): Strong consistency, weak consistency, and eventual consistency

Understanding Isolation

1. Read Uncommitted: The transaction will not lock when reading data to avoid dirty reads and non-repeatable reads. For example, transaction 1 can update data, but

transaction 2 can read only the uncommitted data of transaction 1.

2. Read Committed: When we modify data in a transaction if the transaction has not been committed, other transactions cannot read the data, or can only read the committed data. In the words, the read operation is only locked at the moment of reading and the lock will be released after the end of the reading. This can avoid dirty reads but cannot avoid non-repeatable reads.

3. Repeatable Read: this can avoid the non-repeatable reads. It is locked at the moment of reading, but the lock is not released after the end of the reading and will be released until the end of the transaction.

4. Serialization: It can solve all problems of consistency completely. It is solved by adding table locks.