[Collapse](https://cyberactive.bellevue.edu/webapps/discussionboard/do/message?action=list_messages&course_id=_518940_1&nav=discussion_board&conf_id=_759108_1&forum_id=_2494243_1&message_id=_46532169_1)

Top of Form

Consistency in databases is ensuring that all the copies of the same data on different instances of a server (or on different servers) is synchronized. This ensures that all clients/users see the same thing when querying a database either through the web or some other application. Consistency guarantees that the database doesn't present data anomalies or contradictions (Alastairn, 2023) .

Update consistency ensures that when multiple clients or processes concurrently update the same data item, the updates do not contradict. Update consistency is important to prevent conflicts and maintain data integrity.

Read consistency refers to the consistency of read operations in a distributed system. It ensures that when a client or user reads data, they see the most recent and consistent version of that data. Read consistency is important because it provides accurate and reliable information to users, enabling them to make informed decisions based on up-to-date data. Relational databases can ensure read consistency with ACID transactions. NoSQL databases can sometimes use transactions, and sometime they use atomic aggregate updates (Sadalage et al., 2013) .

Write-write conflicts occur in distributed systems when multiple clients or processes attempt to update the same data item simultaneously. This can lead to conflicts where one update overwrites the changes made by another update, resulting in data inconsistencies and lost updates. This issue can be managed with a pessimistic approach or an optimistic approach. With the pessimistic approach there is some sort of logical lock that only allows one update to be processed, and other updates will not be able to be input. With an optimistic approach will detect the inconsistency and then possibly allow or disallow the update (Sadalage et al., 2013) .

Read-write conflicts occur when a client or process tries to read data while another client or process is simultaneously updating the same data item. In such cases, the reader may see an inconsistent state of the data if the read occurs before the update is applied. Read-write conflicts can lead to incorrect or outdated information being read, which can have adverse effects on the application or system relying on the data. Managing read-write conflicts is important to ensure that readers always see a consistent and up-to-date view of the data.

Alastairn. (2023, February 15). *Database consistency*. ScyllaDB. <https://www.scylladb.com/glossary/database-consistency/>

Sadalage, P. J., & Fowler, M. (2013). Chapter 5. Consistency. In *NoSQL distilled a brief guide to the Emerging World of Polyglot Persistence*. essay, Pearson Education.

Bottom of Form