1. In the context of NoSQL databases, what is consistency and why is it important?

The *consistency* property of a database means that once data is written to a database successfully, queries that follow are able to access the data and get a consistent view of the data. In practice, this means that if you write a record to a database and then immediately request that record, you’re guaranteed to see it. It’s particularly useful for things like Amazon orders and bank transfers.

However, in the NoSQL world, consistency generally falls into one of two camps:

* ACID Consistency (ACID stands for Atomicity, Consistency, Isolation, Durability): ACID means that once data is written, you have full consistency in reads.
* Eventual Consistency (BASE): BASE means that once data is written, it will eventually appear for reading.

Fowler, Adam. APPLYING CONSISTENCY METHODS IN NOSQL. Found on <https://www.dummies.com/programming/big-data/applying-consistency-methods-in-nosql/>

Sourced on 8 May, 2019.

1. What is update consistency and why is it important?

A basic in-house rule to follow when working to update your database, how it will be updated, the parameters to invoke, etc.

1. What is read consistency and why is it important?

Queries in them must produce consistent data for the duration of the transaction, not reflecting changes by other transactions. You can achieve transaction-level read consistency by using explicit locking, read-only transactions, serializable transactions, or by overriding default locking.

This means that the data seen by all queries within the same transaction is consistent with respect to a single point in time, except that queries made by a serializable transaction do see changes made by the transaction itself. Transaction-level read consistency produces repeatable reads and does not expose a query to phantoms.

Found on <https://docs.oracle.com/cd/B13789_01/server.101/b10743/consist.htm>

Sourced on 8 May, 2019

1. What are write-write conflicts?

When two people are updating data at the same time using different formats//formations. Essentially, one of the updates will be over written and would be considered a “lost update”

1. What are read-write conflicts?

The order of events when a table is being read//referenced. If one user if referencing a table while another user is updating on another, the data will not be read until the update is complete and most likely user 1 will no read that data unless notified of the update.