**IN THE CONTEXT OF NOSQL DATABASES, WHAT IS CONSISTENCY AND WHY IS IT IMPORTANT?**

A distributed system keeps copies of the same data on various machines in order to support high availability and scalability. When an application modifies the data on one machine, that change does not immediately propagate to all the database system copies. During the time the change propagates to all replicas, the copies will be as a group, inconsistent.

Simply put, consistency refers to how the data on a distributed system may vary within the same copies of the database system at a given point, usually after a change has been made on one machine and not yet propagated to one of the copies.

Consistency is important because an application might be serving a version of the data that does not reflect the latest update.

**WHAT IS UPDATE CONSISTENCY AND WHY IS IT IMPORTANT?**

In distributed systems update consistency occurs when one node is used as the target for all the updates and then replicating the changes to the other DB system replicas. This avoids write-write conflicts as different copies might have different data at a given point and it would be impossible to check what was updated first and when to propagate what changes to what machines.

**WHAT IS READ CONSISTENCY AND WHY IS IT IMPORTANT?**

Read consistency refers to session consistency where the data that is served by querying the DB system comes from the same node while the session lasts. This is important because it ensures that after a user has made a change to the DB system, he continues to see that change.

**WHAT ARE WRITE-WRITE CONFLICTS?**

When data updates can be made to more than one node – two people might attempt to update a given record at the same time – creating a write-write conflict.

**WHAT ARE READ-WRITE CONFLICTS?**

Read-write conflicts occur when changes made to a DB that have not been yet committed, are overwritten.