

Spring Professional Exam Tutorial v5.0

Question 20

Question 20 - Why is the term "unit of work" so important and why does JDBC AutoCommit violate this pattern?

Unit of work is a generic term to describe, set of tasks that are performing some changes on the data, with assumption that all changes needs to be performed, or no changes should be performed at all.

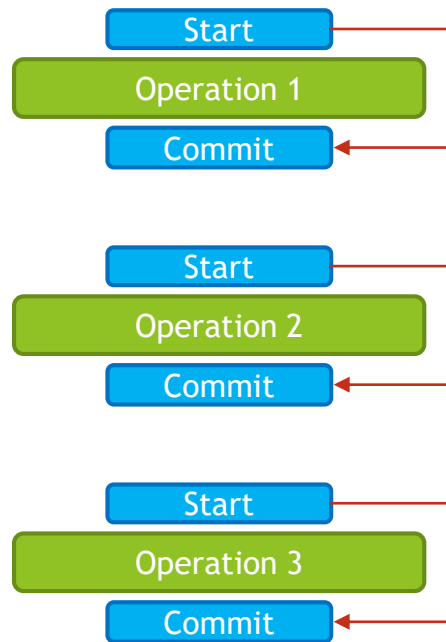
In Relational Databases, Unit of Work can be represented by Database Transaction, which Atomic nature describes “all-or-nothing” behavior described above.

In context of JPA/Hibernate, Unit of Work tracks all changes made to the Data Objects representing entries in the database, and once done, ORM figures out all changes that needs to be applied to the database. This way amount of calls to the database can be minimized by aggregating all changes into one call.

Question 20 - Why is the term "unit of work" so important and why does JDBC AutoCommit violate this pattern?

JDBC AutoCommit violates Unit of Work, because it makes every SQL statement being invoked in a separate transaction that is committed after SQL is executed, this makes impossible to implement Unit of Work consisting of multiple SQL operations.

JDBC with AutoCommit



JDBC without AutoCommit

