JDBC Template

JDBC API is low level api. We need to handle all level requirement
Such as creating/closing connection/statement, exception handling,
resource cleaning, transaction management-rollback/commit... Etc

Spring JDBC Template uses Template pattern which define algorithm
as set of step lets subclass or user implement couple of steps.

In Spring JDBC, we only care about definining connection parameters
or getting from JNDI, setting query and mapping result. That's it!

Spring JDBC template decouples client from low level programming.

Problem: Writing boiler plate code is open for errors.
In case exception, Developer doesnt know what to do.
If the SQLException is not handled, it will propogate to other exception
Unrelated method. It will make coupling between the methods.

Solution: Spring handles all boiler plate code which is low level
programming and exception handling

In Service layer we have core business logic. In Data
Access layer we define data access business logic.

With JDBC template and DI pattern we decouple
Business logic from Data access logic.