**Transaction Management in Spring**

A transaction is a unit of work in which either all operations must execute or none of them.

Example : “Transferring Amount from one account to another “ – this operation includes below at least below two steps

* Deduct the balance from the sender’s account
* Add the amount to the receiver’s account.

Now think of the situation where the amount is deducted from the sender’s account, but not gets delivered to receiver account due to some errors. Such issues are managed by transaction management in which both the steps are performed in a single unit of work where either both steps are performed successfully or in case anyone gets failed, it should be roll backed.

There are four important terms which are very important to understand.

* **Atomic -**Atomicity makes sure that either all operations within a transaction must be successful or none of them.
* **Consistent**- This property makes sure that data should be in consistent state once the transaction is completed.
* **Isolated-**  this property allows multiple users to access the same set of data and each user’s processing should be isolated from others.
* **Durable –**The changes of a successful transaction occurs even if the system failure occurs.

