SpringTransactions:-

\*\* Before Understanding of the spring Transactions we have to see what is Transactions and how many types of transactions and how can we achieve Transactions with springs

Transactions are nothing but single unit of work .either all resources has to commit or all has to rooback

if you want to achieve this we need some other person help that is nothing but Transactions

\*\*Incase of Local Transactions we can achieve this without using this transaction but writeing that logic is very complicated

con1.commit();

con2.commit();

\*\* Here two different resources we are using and commiting indipendently.so if the exception comes after commiting the first resource and just before commiting the second resource insuch a case first resource will be commited and second resource will throw exeption but here we need to write manually rollback operation for first resource as well as second resource if first resource throws exception.

\*\* Incase of normal application by using servlets (Controllers) and Jsp(View) and for business logic everything we will just use pojo classes.This is the simple architecture for simple servlets based applications

\*\* If you are perfoming local transactions using JDBC API.this api it self provided TransactionManager and some more interfaces as part of JDBC Local Transactions.But Jdbc Api will not allo to perform GlobalTransactions

\*\*For this you need to have ApplicationServer which will implement JTA and JDBC 2.0 API with XA Driver implementation or JPA or Hibernate MyBatties or e.t.c

\*\*Incase of local Transactions The data will be commited or rollbacked from only one resource and this will be done by using one phase commit

\*\*But incase of Global Transactions we have to use either of above stated technologies.and i have devloped using Weblogic Application Server.There i have configured two different DataSources and also called ResourcesManages.Those are MySql ResourceManager and OracleResourceManager

\*\*And you have to perform the lookup operation for UserTransactionInterface which is comming from JTA so it will support Global Transactions unless like JDBC Transactions.This will support Two phase commit and i have explained how to perform two phase commit operation in phae commit in JavaGlobalTransactionsWithTwoPhaseCommit in servlets Projects.And the possibilites provideing by weblogic servers to users are also explained

\*\* If you have XA Driver and if you dont have XA Driver What will happen. what are the possibilites

\*\* But incase of Struts kind of frame work application devlopment managing transactions are easy i have explined her only local transactions.

\*\*In struts based frame work bussiness Deligates will talks with the Dao classes.Bussiness Deligates will

call Dao classes by passing Bo objects for persisting the data.what will happen if you dont apply the transactions there each dao class will create the connections and perform the rollback and commit inside their respective classes.

\*\* Insuch case my delegate is responsible for performing the Transactions.before calling Dao class persistence methods my Deligate class has to begian the transactions so transaction manager will enlist the all the resources which are participating in the active transaction

\*\*Transaction Manger will take the responsibility of rollbacking and commiting the activetransaction.we should not give this to any of the dao classes.this responsibility is assosiated with deligate classes

\*\* In this way Struts transactions will work.

\*\*In case of Spring Transaction it is very easy every thing will be taken care by Spring IOC Container

\*\*The same way of Structs will not let Daos to commit and rollback the transactions.if it the case inconsistency results will occur

\*\*We have to write this logic inside the service class.there are many service classes are there in each and every class we have to perform the Transaction .so code will be duplicated.

\*\*inside the service class we will write beginTransaction and commit transaction and rollback the transaction when ever it is the exception by persistence class and reported by service class

\*\*It seems like cross cutting logic that we can apply to every service class.this logic will be required by every devlopers.so it seems like boliler plate logic that every devloper has to write so spring it self provided this

\*\* The class is written here the same way spring internally provided this advice class. incase of declarative approach. we have to apply this advice class to target class. through declarative approach by configuring it in spring bean configuration file

\*\*Incase of annotations how we will apply, that example also explained in annotation approach

\*\*The advantage of this Spring transactions is it will supprort multiple transaction technologies

\*\* you can use any persistence logic and any transaction technology logic spring transactions will support all the technologies

\*\*In case of other frame works like struts if you want to modify transaction logic or persistence logic you need to modify the dao classes with respect that technology

\*But if you use any persistence technology like jpa , spring Transactions , jta , hibernate those specific exceptions will be converted into spring jdbc exception so you no need to write any technology specific exceptions. you can avoid this if you are using this

\*\* Spring Transactions providing Some classes in parllel to Respective Transaction Technologies

\*\*if you are using local transactions you have to use spring provided DataSourceTransactionManager and for Jta, Jpa ,hibernate mybaties also they provided specific implementation classes

\*\*So you no need to write any technology specific implemenetation logic here internally spring will providing this. so you no need to rewrite any technology specific implementation in service classes of your spring application

\*\*Bcz we are not writing the begin transaction and commiting and rollbacking here internally spring transaction aop advice class will take care of these things.for all transactionManagres one common interface is there i.e Transactions.

\*\*if you are using Local Transaction you will configure DataSource Transaction Manager this will be injected to That Advice class this class object will be hold in That interface they designed to Interface

\*\*Incase of Struts kind of frame work applications you can use any technology for transactions either it may be Local Transactions or jpa or jta or hibernate ther bussiness delegates are thightly coupled with the technology specific Transactions.if you are changing from one technology to another technology we have to rewrite those classes

\*\*But if you go for SpringTransactions it is providing the Abstraction over those technologies it will internally talks with those technologies you no need to write any code in service class.you just have to tell to the spring use this technology for transaction by configuring it in spring Bean Configuration file or using java Config file

//This is SpringTransactionAround Advice class(Test implementation)

class TransactionAdvicer implemets MethodInterceptor ,ThrowsAdvice{

private TranactionManager transactionManager;

@Override

public object invoke(MethodInvocation methodInvocation){

transactionManager.begin();

object ret = methodInvocation.proceed();

transactionManager.commit();

}

public void afterThrowing(Exception e){

transactionManager.rollback();

}

}

Spring Transaction Attributes:-

\*\* These are use for specifying the methods you want to advice this aspect and used to configure checked exceptions for rollbacking the operation.by default Spring Transactions will not perform the rollback operation for Checked exceptions

------

At the time of Spring Transactions inernally SpringTransaction Manager will use Thread Local Object to identify the Connection Object.

\*\*Bcz first Dao got one connection object the same connection object has to use for the second Dao in that case some times there is a mismatch of connection objects in Dao in that case Transaction Manager will use the Thread Local.it will Store the current Connection Object inside that thread and the same will be used for the second dao also

What is isolation and Propagation?

Before understanding Isolation we have understand the problems while retriveing the data from Database

1) Dirty read

\*\* Dirty read occurs when one transaction is changing the record, and the other transaction can read this record before the first transaction has been committed or rolled back. This is known as a dirty read scenario because there is always the possibility that the first transaction may rollback the change, resulting in the second transaction having read an invalid data

2) Non-Repeatable read

\*\* Non Repeatable Reads happen when in a same transaction same query yields to a different result. This occurs when one transaction repeatedly retrieves the data, while a difference transactions alters the underlying data. This causes the different or non-repeatable results to be read by the first transaction.

3) Phantom read

\*\* Phantom read occurs where in a transaction execute same query more than once, and the second transaction result set includes rows that were not visible in the first result set. This is caused by another transaction inserting new rows between the execution of the two queries. This is similar to a non-repeatable read, except that the number of rows is changed either by insertion or by deletion.

\*\* This we can avoid by using the isolation.We have to tell to Database while reading thhae data from active transaction dont provide the uncommited data.so that we can avoid Dirty Read problem

\*\*While querying the data from database inside the active transaction no other user not allowed to commit the transaction.So that we can avoid the Non Repeated problem

If you apply this Dirty read problem automatically will be solved it is supereior than this.but the problem is that performence problem

\*\*These can be achieved by isolation levels. every database vendor will provide supprt for this problems

oracle people provided 4 Types of isolation levels.mysql and ms server also provided the support for this

1)READ UNCOMMITED

2)READ COMMITED( By default this operation will perform)

3)REPEATABLE\_READ

4) NON REPEATABLE\_READ

\*\* Serializable this superior to all the isolation levels.this will lock the all the resources read amd write e.t.c