Transaction transaction = session.beginTransaction();

We need the Transaction object only when we are performing the database operation, so unless we have Session object we dont perform db operations, so let the Session acts as an Factory for instantiating the TransactionFactory implementation based on the hibernate.transaction.factory_class property in hibernate.cfg.xml

In our application we need to instantiate the TransactionFactory implementation class depends on the type of Transaction and call getTransaction() method that returns the implementation of Transaction. using the Transaction implementation we can commit/rollback the transaction

```
interface TransactionFactory {
    Transaction getTransaction();
}
```

```
class
JdbcResourceLocalTransactionFactoryImpl
implements TransactionFactory{}
```

```
class JtaTransactionFactoryImpl implements
TransactionFactory {}
```

creates the object of JdbcResourceLocalTransaction by wrapping Connection object into it

```
interface Transaction {
    void commit();
    void rollback();
    ….
}
```

reads the hibernate.cfg.xml for jndiRegistry properties and performs a lookup into appServer in getting the userTransaction object and wraps into JtaTransactionImpl object and returns it

We need to instantiate the object of JdbcResourceLocalTransactionImpl by wrapping Connection object into it

```
class
JdbcResourceLocalTransactionImpl
implements Transaction {}
```

```
class JtaTransactionImpl implements
Transaction {}
```

We need to create the object of JtaTransactionImpl by wrapping userTransaction object by pulling it from jndiRegistry of the application server

jdbc: local
con: [commit/rollback]

jta: global
userTransactionManager: [commit/rollback]

database

A) upon calling session.beginTransaction(), the Session object goes to the hibernate.cfg.xml and looks for transaction.factory_class and picks the corresponding implementation class of the TransactionFactory
1. if factory_class='jdbc' then it instantiates JdbcResourceLocalTransactionFactoryImpl
2. if factory_class='jta' then it instantiates the JtaTransactionFactoryImpl

B)then on that Factory, it invokes getTransaction() method and the returned implementation of Transaction interface, it invokes begin() transaction and returns the object to us.

C) At the end of the operation, within the finally block we need to commit/rollback the transaction