**components of Hibernate**

**\*\*configuration--->**its class

\*public Configuration **addClass**(Class persistentClass)throws MappingException

persistentClass - The mapped class

\*public **SessionFactory buildSessionFactory()** throws HibernateException

Instantiate a new SessionFactory, using the properties and mappings in this configuration.

The SessionFactory will be immutable,

so changes made to the Configuration after building the SessionFactory will not affect it.

\*public Configuration **configure()**throws HibernateException

\*public Configuration **configure(String resource)**throws HibernateException

\*public Configuration **configure(URL url)**throws HibernateException

\*public Configuration **configure(File configFile)** throws HibernateException

**\*\*\*SessionFactory----->its Interface**

\*Session **openSession()**throws HibernateException

\*Session **openSession(Interceptor interceptor)**throws HibernateException

\*Session **openSession(Connection connection)**

\*Session **openSession(Connection connection,Interceptor interceptor)**

\*void **close()**throws HibernateException

Destroy this SessionFactory and release all resources (caches, connection pools, etc).

It is the responsibility of the application to ensure that there are no open sessions

before calling this method as the impact on those sessions is indeterminate.

No-ops if already closed.

**\*\*\*\*Session----->interface**

\*Transaction **beginTransaction()**throws HibernateException

Begin a unit of work and return the associated Transaction object. If a new underlying

transaction is required, begin the transaction. Otherwise continue the new work in the

context of the existing underlying transaction. The class of the

returned Transaction object is determined by the property hibernate.transaction\_factory.

\*Transaction **getTransaction()**

\*void **update(Object object)** throws HibernateException

\*void **update(String entityName,Object object)**throws HibernateException

\*Serializable **save(Object object)**throws HibernateException

\*Serializable **save(String entityName,Object object)**throws HibernateException

\*Object **load(Class theClass, Serializable id)**throws HibernateException

\*Object **load(String entityName,Serializable id)**throws HibernateException

\*void **load(Object object,Serializable id)**throws HibernateException

\*Object **load(String entityName,Serializable id,LockOptions lockOptions)**throws HibernateException

\*Object **load(Class theClass,Serializable id, LockMode lockMode)**throws HibernateException

\*Object **load(Class theClass,Serializable id,LockOptions lockOptions)**throws HibernateException

\*Object **get(Class clazz,Serializable id)**throws HibernateException

\*Object **get(Class clazz,Serializable id,LockMode lockMode)**throws HibernateException

\*Object **get(Class clazz,Serializable id,LockOptions lockOptions)**throws HibernateException

\*Object **get(String entityName,Serializable id)**throws HibernateException

\*Object **get(String entityName,Serializable id,LockMode lockMode)**throws HibernateException

\*Object **get(String entityName,Serializable id,LockOptions lockOptions**) throws HibernateException

\***Connection close()**throws HibernateException

\*void **delete(Object object)**throws HibernateException

\*void **delete(String entityName,Object object)**throws HibernateException

**\*\*\*\*transaction**

\*void **rollback()**throws HibernateException

\*void **commit()**throws HibernateException

**\*\*\*\*\*\*QUERY**

\*Query **createQuery(String queryString)**throws HibernateException

\*SQLQuery **createSQLQuery(String queryString)**throws HibernateException

\*Criteria **createCriteria(Class persistentClass)**

\*Criteria **createCriteria(Class persistentClass,String alias)**

\*Criteria **createCriteria(String entityName)**

\*Criteria **createCriteria(String entityName,String alias)**