Hibernate Architecture

In this Chapter, we will discuss High level building blocks of Hibernate. From a very High level, we can consider Hibernate as a black box which takes application data in an Object form and stores them in Relational Database and vice versa. It supports all variants if RDBMS like mysql,oracle,DB2 etc.

If we dig down , we can see Hibernate offers six main components which make developers life easy, in this article we discuss those High-Level components. Hibernate implements layered architecture and encapsulate each component perfectly.



**Hibernate Architecture**

**High Level Components**

1. **Configuration :** To create a Hibernate Application this is the first Object developer has to initialize. Configuration Object holds two main things a. Database related information i.e schema name,user,password,connection URL etc and b. Mapping resources basically the Domain Object or Persistent Object mapping so Hibernate can understand which Object maps to which table.

Configuration Object takes this values from the hibernate.xml or hibernate.properties file, by doing this Hibernate make a flexible design so a user can change the parameter on the fly without a change in source code. One way we can tell Hibernate separates the Strategy part or definition part from Implementation. A nice way to achieve abstraction.

**2. Session Factory :** From Configuration Object we create the SessionFactory. It can be thought as an Application context of Hibernate Application. Session Factory Object is very Costly so it is better to create it at Application start up, As the name suggest Sessionfactory manages/creates all the Sessions. Session factory is thread Safe so multiple threads can act on Sessionfactory without dirtying its state.

One key point to remember There is only one SessionFactory for Only one underlying Database , so if an Application has multiple databases It has to maintain multiple SessionFactory.

**3. Session:** Session Object is the heart of Hibernate. Every Operation is associated with Session like insert,update,delete. Session Object is created from SessionFactory and it is a lightweight/noncostly Object.Hibernate talks with Database using Session, also Hibernate maintains a First Level cache in Session. The session is not thread- safe so it is advisable that not open session for a long time unless data can be dirty if we work in a multithreaded environment where multiple threads are involved with the same session.

**4. Transaction :** We can create Transaction from Session Object. The transaction represents a unit of work and maintains ACID property. Hibernate uses underlying Transaction Manager to handle the transaction. It can be JDBC or JTA. There are two types of Transaction

**CMT** : ***Container Managed Transaction***, It is a coarse-grained transaction where container handles the transaction for you developer does not have to bother about the transaction. Here JDBC or JTA takes the responsibility.

**BMT**: ***Bean Managed Transaction***, In a few cases, we need fine grained control over the transaction in that case we don’t delegate the transaction handling to Container Transaction managers we manage it programmatically.

**5. Query:** In Hibernate we use Query API to fetch data from the database. There are two types of query we can use

**a. HQL** : Hibernate Query Language , which is a kind of Domain Specific language and it acts upon Domain Objects/Persistent Objects. The best thing about HQL is ,it’s syntax is same as SQL so the learning curve is very less and it is acting on Objects rather than the table

**b. SQL** : You can directly pass SQL query into Query API.

**6. Criteria** : Hibernate provide Criteria API by using this you can create dynamic queries. If you don’t know the input parameters of the query but it populates at runtime then Criteria is the best choice. Say for building Search Query Criteria is a good choice.