[**Hibernate五大对象**](http://blog.csdn.net/ee470474137/article/details/45744269)

所有的Hibernate应用都会访问它的5个核心接口，分别如下：   
Configuration接口：   
SessionFactory接口：   
Session接口：   
Transaction接口：   
Query和Criteria接口：

1. **Configuration接口 :**  
   Configuration用于配置并启动Hibernate。Hibernate应用通过Configuration的实例来指定对象-关系映射文件，或通过Configuration动态配置Hibernate的属性，然后通过Configuration来创建相应的SessionFactory实例。Configuration相当于hibernate上下文。
2. **SessionFactory接口 :**  
   一个SessionFactory对应一个数据源，它是个重量级对象，不可随意生成多个实例。对于一般的单[**数据库**](http://lib.csdn.net/base/mysql)应用来说，只需要一个SessionFactory就足够了。当然如果有多个数据库的话，还是需要为每个数据库生成对应的SessionFactory。它是线程安全的，同一个实例可以被应用中的多个线程共享。   
   也许你会很好奇，SessionFactory为什么是重量级对象呢？我也同样好奇，通过查看Hibernate的源码，发现SessionFactory存放了大量预定义的SQL语句以及映射元数据，所以自然需要很大的缓存了，同时需要一定的CPU时间来计算生成。想想Hibernate的这个设计是很有意义的，因为有了Mapping文件，很多SQL语句就已经确定了，只需要动态生成一次就可以了，这个设计也是为了提高持久化的效率。**SessionFactory负责产生session对象。是hibernate的二级缓存。**
3. **Session接口 ：**  
   从SessionFactory中可以获得Session实例。   
   Session接口是Hibernate应用中使用最广泛的接口了，它是持久化管理器，提供添加、更新、删除、加载、查询对象。Session不是线程安全的，所以应避免多个线程共享同一个Session实例。Session是轻量级对象，它的创建和销毁不需要太多资源，这意味着在应用中可以经常创建和销毁Session对象。   
   Session有一个缓存，称之为Hibernate的一级缓存，它存放当前工作单元加载的持久化对象，每个Session都有自己的缓存，缓存中的对象只能被当前工作单元访问。**Session级别的缓存意义不是很大。**

**4、Transaction接口：**  
Transaction是Hibernate的数据库事务接口，它对底层DAO事务接口进行了封装，底层事务接口包括:   
JDBC API   
JTA([Java](http://lib.csdn.net/base/java)Transaction API)   
CORBA(Common Object Requet Broker Architecture) API   
Hibernate应用可以通过一致Transaction接口来声明事务边界，这有助于应用可以在不同的环境或容器中移植。具体的事务实现使用在Hibernate.properties中进行指定。

**5、Query和Criteria接口 ：**  
这两个是Hibernate的查询接口，用于向数据库查询对象，以及控制执行查询的过程。Query实例包装了一个HQL(Hibernate Query Language)来查询。Criteria接口完全封装了基于字符串形式的查询语句，比Query更面向对象，Criteria更擅长执行动态查询。