# java hibernate 面试题

1. hibernate是一种开源的对象关系映射框架，它对[JDBC](https://baike.baidu.com/item/JDBC/485214" \t "https://baike.baidu.com/item/Hibernate/_blank)进行了非常轻量级的对象封装，它将POJO与数据库表建立映射关系，是一个全自动的orm框架，hibernate可以自动生成SQL语句，自动执行。
2. Hibernate用反射机制实现持久化对象操作，实现与[IDE](https://baike.baidu.com/item/IDE/8232086" \t "https://baike.baidu.com/item/Hibernate/_blank)(Integrated Development Environment)的耦合度。

3.Hibernate框架技术最关键是**数据持久化**，是将数据保存到持久层的过程。

4.系统结构的层次模型有三个阶段：

①整个过程首先实现应用层和数据层。数据层保存持久化数据，应用层接收输入的数据。

②然后通过MVC 模式实现业务逻辑与表示层的分开。表示层和用户实现交互，业务逻辑层处理数据持久化操作。

③将第二阶段业务逻辑层的功能部署拆分后，业务逻辑层完成核心业务逻辑处理，持久层完成对象持久化。降低业务逻辑层复杂度的同时将数据持久化让其他组件完成。

1. 特点：

①将对数据库的操作转换为对Java对象的操作，从而简化开发。通过修改一个“持久化”对象的属性从而修改数据库表中对应的记录数据。

②提供线程和进程两个级别的缓存提升应用程序性能。

③有丰富的映射方式将Java对象之间的关系转换为数据库表之间的关系。

④屏蔽不同数据库实现之间的差异。在Hibernate中只需要通过“方言”的形式指定当前使用的数据库，就可以根据底层数据库的实际情况生成适合的SQL语句。

⑤非侵入式：Hibernate不要求持久化类实现任何接口或继承任何类，POJO即可。

1. Hibernate的API一共有6个，分别为:[Session](https://baike.baidu.com/item/Session/479100" \t "https://baike.baidu.com/item/Hibernate/_blank)、[SessionFactory](https://baike.baidu.com/item/SessionFactory" \t "https://baike.baidu.com/item/Hibernate/_blank)、[Transaction](https://baike.baidu.com/item/Transaction/4114110" \t "https://baike.baidu.com/item/Hibernate/_blank)、[Query](https://baike.baidu.com/item/Query" \t "https://baike.baidu.com/item/Hibernate/_blank)、[Criteria](https://baike.baidu.com/item/Criteria" \t "https://baike.baidu.com/item/Hibernate/_blank)和Configuration。通过这些[接口](https://baike.baidu.com/item/%E6%8E%A5%E5%8F%A3" \t "https://baike.baidu.com/item/Hibernate/_blank)，可以对持久化对象进行存取、事务控制。

①Session:负责执行对持久化对象的crud操作，session对象是非线程安全的。

②SessionFactory:负责对hibernate初始化，充当数据存储源的代理，负责创建session对象；非轻量级，一个项目中一个SeesionFactory就可以，如果要操作多个数据库，可以为每个数据库指定一个SeesionFactory。

③Transaction:对实际事物实现的一种抽象，可选接口，对事物的处理。

④Query:对数据库及持久对象进行查询操作，有HQL语言及本地sql语言，Query经常被用来绑定查询参数、限制查询记录数量，并最终执行查询操作。

⑤Criteria：与Query接口非常类似，允许创建并执行面向对象的标准化查询。值得注意的是Criteria接口也是轻量级的，它不能在Session之外使用。

⑥Configuration 类的作用是对Hibernate 进行配置，以及对它进行启动。在Hibernate 的启动过程中，Configuration 类的实例首先定位映射文档的位置，读取这些配置，然后创建一个SessionFactory对象。

1. 包的作用：

**net.sf.hibernate.\*：**该包的类基本上都是接口类和异常类

**net.sf.hibernate.cache.\*：**JCS的实现类

**net.sf.hibernate.cfg.\*：**[配置文件](https://baike.baidu.com/item/%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6" \t "https://baike.baidu.com/item/Hibernate/_blank)读取类

**net.sf.hibernate.collection.\*：**Hibernate集合接口实现类，例如List，Set，Bag等等，Hibernate之所以要自行编写集合接口实现类是为了支持lazy loading

**net.sf.hibernate.connection.\*：**几个[数据库连接池](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E8%BF%9E%E6%8E%A5%E6%B1%A0" \t "https://baike.baidu.com/item/Hibernate/_blank)的Provider

**net.sf.hibernate.dialect.\*：**支持多种数据库特性，每个Dialect实现类代表一种数据库，描述了该数据库支持的数据类型和其它特点，例如是否有AutoIncrement，是否有Sequence，是否有[分页](https://baike.baidu.com/item/%E5%88%86%E9%A1%B5" \t "https://baike.baidu.com/item/Hibernate/_blank)sql等等

**net.sf.hibernate. eg.\*：**Hibernate文档中用到的例子

**net.sf.hibernate.engine.\*：**这个包的类作用比较散

**net.sf.hibernate.expression.\*：**HQL支持的表达式

**net.sf.hibernate.hq.\*：**HQL实现

**net.sf.hibernate. id.\*：**ID生成器

**net.sf.hibernate.impl.\*：**最核心的包，一些重要接口的实现类，如Session，SessionFactory，Query等

**net.sf.hibernate.jca.\*：**[JCA](https://baike.baidu.com/item/JCA" \t "https://baike.baidu.com/item/Hibernate/_blank)支持，把Session包装为支持JCA的接口实现类

**net.sf.hibernate.jmx.\*：**JMX是用来编写App Server的[管理程序](https://baike.baidu.com/item/%E7%AE%A1%E7%90%86%E7%A8%8B%E5%BA%8F" \t "https://baike.baidu.com/item/Hibernate/_blank)的，大概是JMX部分接口的实现，使得App Server可以通过JMX接口管理Hibernate

**net.sf.hibernate.loader.\*：**也是很核心的包，主要是生成sql语句。

**net.sf.hibernate.lob.\*：**Blob和Clob支持

**net.sf.hibernate.mapping.\*：**hbm文件的属性实现

**net.sf.hibernate.metadata.\*：**PO的Meta实现

**net.sf.hibernate.odmg.\*：**ODMG是一个ORM标准，这个包是ODMG标准的实现类

**net.sf.hibernate.persister.\*：**核心包，实现持久对象和表之间的映射

**net.sf.hibernate.proxy.\*：**Proxy和Lazy Loading支持

**net.sf.hibernate. ps.\*：**该包是PreparedStatment Cache

**net.sf.hibernate.sql.\*：**生成JDBC [sql语句](https://baike.baidu.com/item/sql%E8%AF%AD%E5%8F%A5" \t "https://baike.baidu.com/item/Hibernate/_blank)的包

**net.sf.hibernate.test.\*：**测试类，你可以用[junit](https://baike.baidu.com/item/junit" \t "https://baike.baidu.com/item/Hibernate/_blank)来测试Hibernate

**net.sf.hibernate.tool.hbm2ddl.\*：**用hbm[配置文件](https://baike.baidu.com/item/%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6" \t "https://baike.baidu.com/item/Hibernate/_blank)生成DDL

**net.sf.hibernate.transaction.\*：**Hibernate Transaction实现类

**net.sf.hibernate.type.\*：**Hibernate中定义的持久对象的属性的数据类型

**net.sf.hibernate.util.\*：**一些工具类，作用比较散

**net.sf.hibernate.xml.\*：**XML数据绑定

1. Hibernate缓存：

①一级缓存：session级别缓存，事务范围缓存，一般不用人工干预。当应用程序调用Session的save()、update()、saveOrUpdate()、get()或load()，以及调用查询接口的 list()、iterate()或filter()方法时，如果在Session缓存中还不存在相应的对象，Hibernate就会把该对象加入到第一级缓存中。当清理缓存时，Hibernate会根据缓存中对象的状态变化来同步更新数据库。

Session为应用程序提供了两个管理缓存的方法：

evict(Object obj)：从缓存中清除参数指定的持久化对象。 clear()：清空缓存中所有持久化对象。

②二级缓存：SessionFactory级别的缓存，它是属于进程范围或集群范围的缓存，可以进行配置和更改，并且可以动态加载和[卸载](https://baike.baidu.com/item/%E5%8D%B8%E8%BD%BD" \t "https://baike.baidu.com/item/Hibernate/_blank)。

Hibernate的二级缓存策略的一般过程如下：

1) 条件查询的时候，总是发出一条select \* from table\_name where …. （选择所有字段）这样的SQL语句查询数据库，一次获得所有的[数据对象](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AF%B9%E8%B1%A1" \t "https://baike.baidu.com/item/Hibernate/_blank)。

2) 把获得的所有数据对象根据ID放入到第二级缓存中。

3) 当Hibernate根据ID访问数据对象的时候，首先从Session一级缓存中查；查不到，如果配置了[二级缓存](https://baike.baidu.com/item/%E4%BA%8C%E7%BA%A7%E7%BC%93%E5%AD%98" \t "https://baike.baidu.com/item/Hibernate/_blank)，那么从二级缓存中查；查不到，再查询数据库，把结果按照ID放入到缓存。

4) 删除、更新、增加数据的时候，同时更新缓存。

**什么样的数据适合存放到第二级缓存中？**

1 很少被修改的数据

2 不是很重要的数据，允许出现偶尔并发的数据

3 不会被并发访问的数据

4 参考数据,指的是供应用参考的常量数据，它的实例数目有限，它的实例会被许多其他类的实例引用，实例极少或者从来不会被修改。

**不适合存放到第二级缓存的数据？**

1 经常被修改的数据

2 财务数据，绝对不允许出现并发

3 与其他应用共享的数据。

9.**hibernate工作原理：**

1、通过Configuration().configure();读取并解析hibernate.cfg.xml[配置文件](https://baike.baidu.com/item/%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6" \t "https://baike.baidu.com/item/Hibernate/_blank)。

2、由hibernate.cfg.xml中的<mappingresource="com/xx/User.hbm.xml"/>读取解析映射信息。

3、通过config.buildSessionFactory();//得到sessionFactory。

4、sessionFactory.openSession();//得到session。

5、session.beginTransaction();//开启事务。

6、persistent operate;

7、session.getTransaction().commit();//提交事务

8、关闭session;

9、关闭sessionFactory;

**10.hibernate优点：**

1、封装了jdbc，简化了很多重复性代码。

2、简化了DAO层编码工作，使开发更对象化了。

3、移植性好，支持各种数据库，如果换个数据库只要在[配置文件](https://baike.baidu.com/item/%E9%85%8D%E7%BD%AE%E6%96%87%E4%BB%B6" \t "https://baike.baidu.com/item/Hibernate/_blank)中变换配置就可以了，不用改变hibernate代码。

4、支持透明持久化，因为hibernate操作的是纯粹的（pojo）java类，没有实现任何接口，没有侵入性。所以说它是一个轻量级框架。

**11.优化Hibernate**

1、使用一对多的双向关联，尽量从多的一端维护。

2、不要使用一对一，尽量使用多对一。

3、配置对象缓存，不要使用集合缓存。

4、表字段要少，表关联不要怕多，有二级缓存撑腰。

12.**hibernate session 对象存在的三种方式及存在意义**

**瞬时状态**:对象再刚刚被创建出来的时候，刚刚被new出来但是未被session对象里面的方法操作，如果没有使用就会被垃圾回收机制回收；

**持久状态**:对象被session里面的get或者load等方法操作过，并且存在于hibernate的缓存中，对象的数据会随时保存到数据库中

**游离状态**:java对象被session操作完成以后就进入游离状态

13.Hibernate get和load 原理及区别

最直接的区别就是每次使用**get方法**就要从数据库中重新获取，load**方法**是先从hibernate的缓存中获取如果没有然后才从数据库中获取。

1. 一级缓存:session级别的缓存，hibernate自带的，并且一级缓存只存在一次会话中，如果session失效，则缓存被清除。

如果需要清除缓存使用：

session.flush();让一级缓存与数据库同步； 　　　　session.evict();清空一级缓存中指定的对象； 　　　　session.clear();清空一级缓存中所有的对象；