**hibernate缓存机制**

Hibernate提供的缓存有**一级缓存**、**二级缓存**。 目的是为了**减少对数据库的访问次数，提升程序执行效率**！

·一级缓存：

**基于Session的缓存**，缓存内容**只在当前session有效**，session关闭，缓存内容失效！

特点：

**作用范围较小！ 缓存的事件短。**

**缓存效果不明显。**

·二级缓存：

　　Hibernate提供了**基于应用程序级别**的缓存即为二级缓存，可以**跨多个session**，即**不同的session都可以访问缓存数据**。 这个缓存也叫二级缓存。

Hibernate提供的二级缓存有默认的实现，且是一种**可插配的缓存框架**！如果用户想用二级缓存，只需要在hibernate.cfg.xml中**配置即可**；不想用，**直接移除**，不影响代码。

如果用户觉得hibernate提供的框架框架不好用，自己**可以换其他的缓存框架**或自己实现缓存框架都可以。

**二级缓存，使用步骤：**

　　1)开启二级缓存；

　　2)指定缓存框架；

　　3)指定那些类加入二级缓存；

　　4)测试；

**hibernate缓存机制面试题回答方式：**

首先说下Hibernate缓存的作用（即为什么要用缓存机制），然后再具体说说Hibernate中缓存的分类情况，最后可以举个具体的例子。

**Hibernate缓存的作用：**

Hibernate是一个持久层框架，经常访问物理数据库，为了降低应用程序对物理数据源访问的频次，从而提高应用程序的运行性能。缓存内的数据是对物理数据源中的数据的复制，应用程序在运行时从缓存读写数据，在特定的时刻或事件会同步缓存和物理数据源的数据**Hibernate缓存分类：**

Hibernate缓存包括两大类：Hibernate一级缓存和Hibernate二级缓存

Hibernate一级缓存又称为“Session的缓存”，它是内置的，不能被卸载（不能被卸载的意思就是这种缓存不具有可选性，必须有的功能，不可以取消session缓存）。由于Session对象的生命周期通常对应一个数据库事务或者一个应用事务，因此它的缓存是事务范围的缓存。第一级缓存是必需的，不允许而且事实上也无法卸除。在第一级缓存中，持久化类的每个实例都具有唯一的OID。

Hibernate二级缓存又称为“SessionFactory的缓存”，由于SessionFactory对象的生命周期和应用程序的整个过程对应，因此Hibernate二级缓存是进程范围或者集群范围的缓存，有可能出现并发问题，因此需要采用适当的并发访问策略，该策略为被缓存的数据提供了事务隔离级别。第二级缓存是可选的，是一个可配置的插件，在默认情况下，SessionFactory不会启用这个插件。

**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 常量数据

**不适合存放到第二级缓存的数据？**

1 经常被修改的数据

2 .绝对不允许出现并发访问的数据，如财务数据，绝对不允许出现并发

3 与其他应用共享的数据。

**Hibernate查找对象如何应用缓存？**

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，那么从二级缓存中查；如果都查不到，再查询数据库，把结果按照ID放入到缓存,删除、更新、增加数据的时候，同时更新缓存。Hibernate管理缓存实例无论何时，当你给save()、update()或saveOrUpdate()方法传递一个对象时，或使用load()、 get()、list()、iterate() 或scroll()方法获得一个对象时, 该对象都将被加入到Session的内部缓存中。 当随后flush()方法被调用时，对象的状态会和数据库取得同步。 如果你不希望此同步操作发生，或者你正处理大量对象、需要对有效管理内存时，你可以调用evict() 方法，从一级缓存中去掉这些对象及其集合。