Hibernate 面试题分析

1. Hibernate 的检索方式有哪些？

① 导航对象图检索
② OID 检索
③ HQL 检索
④ QBC 检索
⑤ 本地 SQL 检索

2. 在 Hibernate 中 Java 对象的状态有哪些 ？

①. 临时状态（transient）：不处于 Session 的缓存中。OID 为 null 或等于 id 的 unsaved-value 属性值
②. 持久化状态（persistent）：加入到 Session 的缓存中。
③. 游离状态（detached）：已经被持久化，但不再处于 Session 的缓存中。

3. Session 的清理和清空有什么区别？

清理缓存调用的是 session.flush() 方法 . 而清空调用的是 session.clear() 方法.

Session 清理缓存是指按照缓存中对象的状态的变化来同步更新数据库，但不清空缓存；清空是把 Session 的缓存置空, 但不同步更新数据库；

4. load()和 get()的区别

①：如果数据库中，没有 OID 指定的对象。通过 get 方法加载，则返回的是一个 null；通过 load 加载，则返回一个代理对象，如果后面代码如果调用对象的某个属性会抛出异常：org.hibernate.ObjectNotFoundException；
②：load 支持延迟加载，get 不支持延迟加载。

5. hibernate 优缺点

①. 优点:
> 对 JDBC 访问数据库的代码做了封装，简化了数据访问层繁琐的重复性代码
> 映射的灵活性, 它支持各种关系数据库, 从一对一到多对多的各种复杂关系.
> 非侵入性、移植性会好
> 缓存机制: 提供一级缓存和二级缓存

②. 缺点:

> 无法对 SQL 进行优化
> 框架中使用 ORM 原则, 导致配置过于复杂
> 执行效率和原生的 JDBC 相比偏差: 特别是在批量数据处理的时候
> 不支持批量修改、删除

6. 描述使用 Hibernate 进行大批量更新的经验.

直接通过 JDBC API 执行相关的 SQl 语句或调用相关的存储过程是最佳的方式

7. Hibernate 的 OpenSessionView 问题

①. 用于解决懒加载异常, 主要功能就是把 Hibernate Session 和一个请求的线程绑定在一起, 直到页面完整输出, 这样就可以保证页面读取数据的时候 Session 一直是开启的状态, 如果去获取延迟加载对象也不会报错。

②. 问题: 如果在业务处理阶段大批量处理数据, 有可能导致一级缓存里的对象占用内存过多导致内存溢出, 另外一个是连接问题: Session 和数据库 Connection 是绑定在一起的, 如果业务处理缓慢也会导致数据库连接得不到及时的释放, 造成连接池连接不够. 所以在并发量较大的项目中不建议使用此种方式, 可以考虑使用迫切左外连接 (LEFT OUTER JOIN FETCH) 或手工对关联的对象进行初始化.

③. 配置 Filter 的时候要放在 Struts2 过滤器的前面, 因为它要页面完全显示完后再退出.

8. Hibernate 中 getCurrentSession() 和 openSession() 的区别？

①. getCurrentSession() 它会先查看当前线程中是否绑定了 Session, 如果有则直接返回, 如果没有再创建. 而 openSession() 则是直接 new 一个新的 Session 并返回。

②. 使用 ThreadLocal 来实现线程 Session 的隔离。

③. getCurrentSession() 在事务提交的时候会自动关闭 Session, 而 openSession() 需要手动关闭.

9.如何调用原生 SQL？

调用 Session 的 doWork() 方法.

10. 说说 Hibernate 的缓存：

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

1）. Hibernate 一级缓存又称为“Session 的缓存”，它是内置的，不能被卸载。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。在第一级缓存中，持久化类的每个实例都具有唯一的 OID。

2）. 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。

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。