**2023-2024学年第2学期**

**对象关系映射技术作业**

**20232160A0934张嘉辉**

**1．谈谈你对Git的理解,使用Git进行代码提交和代码拉取的过程，如何解决代码提交中的冲突?**

Git 是一个分布式版本控制系统，用于跟踪和管理代码的变更，它允许多人协同工作，记录每次代码修改，并在需要时回溯到特定版本；当出现冲突时，Git 一般会提示哪些文件发生了冲突，使用 git status 查看冲突的文件，手动编辑这些文件，将冲突部分修改为所需的代码。但是这是使用手动编辑的方式，还可以使用ide来可视化解决冲突问题，选择是否合并或者修改。

**2.Hibernate框架中映射关系配置文件的作用是什么、限制条件有哪些、配置的内容有哪些?**

配置文件通常以 .hbm.xml 后缀命名，用于定义 Java 实体类与数据库表之间的映射关系，通过配置映射关系，Hibernate 在启动项目时会自动生成相应的数据库表；映射文件中的类主键和属性必须与数据库表一一对应；类和表的映射、主键的映射、属性和字段的映射、关系的映射。

**3.Hibernate框架中核心配置文件的作用是什么、限制条件有哪些、配置的内容有哪些?**

hibernate.cfg.xml 被称为 Hibernate 的核心配置文件，它包含了数据库连接的相关信息以及映射文件的基本信息；通常情况下，该配置文件默认放在项目的 src 目录下；驱动类、URL 路径、用户名密码、方言、是否显示sql、格式化sql、自动建表

**4.Hibernate框架的核心配置文件中为什么要配置mapping标签引入映射配置文件?**

配置 <mapping> 标签的目的是引入映射配置文件，让 Hibernate 知道如何将 Java 实体类与数据库表之间建立映射关系

**5.谈谈Hibernate的执行过程。**

初始化 Hibernate，构建 Configuration 实例；创建 SessionFactory 实例；创建 Session 实例，建立数据库连接；创建 Transaction 实例，开启一个事务；利用 Session 接口进行持久化操作；提交事务；关闭 Session，断开与数据库连接

**2024.3.23**

**1.使用Session对象进行数据库表的修改或删除操作有什么注意的地方。**

**更新操作**

1、对象状态：

对象必须处于持久化状态才能更新。如果对象是临时对象（瞬时态），你需要先调用session.save()或session.merge()将其转变为持久态。如果你已经修改了一个持久化对象的属性，但在另一个Session中加载了相同实体的副本（即产生了游离态对象），你需要先调用session.update(object)或者session.merge(object)，确保新值被发送到数据库。

2、事务处理：

在调用session.update()之后，确保这些更改在一个事务中提交（transaction.commit()），否则更改不会持久化到数据库。

3、并发控制：

避免并发冲突，尤其是在高并发环境下，可能需要处理乐观锁定问题，如版本字段的自动更新。

4、脏检查：

Hibernate通常仅当检测到对象属性发生变化时才会执行实际的SQL UPDATE操作。因此，如果你直接修改了对象属性而未通过Session更新，可能会导致Hibernate无法识别变化并更新数据库。

**删除操作**

1、对象引用：

要删除一个对象，先要通过session.load()或session.get()获取其持久化实例，然后调用session.delete(object)。

注意，删除后不要再继续使用此对象，因为它的状态不再是可管理的，并且在事务提交后从数据库中移除。

2、级联删除：

如果有相关的关联对象设置了级联删除（cascade），则删除主对象时，相关联的对象也会自动删除。务必了解和配置好级联行为以避免意外的数据丢失。

3、事务边界：

同样，删除操作应在事务内完成，并且只有当事务成功提交时，删除操作才会真正生效。

4、清理缓存：

删除操作后，确保Session缓存中的相关信息得到更新，避免缓存与数据库数据不一致的问题。

**2.在Hibernate中对象有哪些状态，分别有什么特点。**

**瞬时态**：新建但尚未与Hibernate Session关联的对象处于瞬时态。这类对象没有数据库标识符（ID），即没有与数据库中的任何记录关联。瞬时对象的生命周期完全受控于Java堆内存，若无引用指向它，将被Java垃圾回收器回收。瞬时对象的任何改动都不会影响数据库。

**持久态**：与当前Session关联的对象处于持久态。持久对象拥有一个数据库生成或用户指定的唯一标识符（ID）。持久对象的改动会自动同步到数据库中，当事务提交时，Hibernate会根据对象的状态执行相应的INSERT、UPDATE或DELETE SQL语句。只要对象仍处于Session的有效期内，即使Session关闭后再重新打开，Hibernate也能识别出该对象的持久性。

**脱管态/游离态：**曾经是持久态的对象，但现在与任何Session都没有关联，这种状态被称为脱管态或游离态。脱管对象依然保留了与数据库记录关联的标识符（ID），但它不再受到Hibernate的跟踪和管理，其状态改变不会自动反映到数据库。脱管对象可以通过Session.merge()方法重新与Session关联，从而转换回持久态，此时对象的最新状态会被同步至数据库。

**3.谈谈你对缓存的理解。**

缓存是一种用于提高数据访问速度的存储技术，它的基本原理是在内存中存储一份常用数据的副本，以便快速响应后续的相同请求，减少对原始数据源（如硬盘、数据库等）的访问次数，从而提升系统性能并降低延迟。

**4.对于持久态的对象，hibernate中的缓存有什么功能，并说明该功能的原理。**

**数据缓存**：当我们在当前Session中通过save(), update() 或 load()等操作得到一个持久态对象时，该对象会被缓存到Session缓存中。后续对同一对象的引用将直接从缓存中获取，而不是每次都去数据库查询，从而极大地提高了数据访问效率。

**脏数据检查与同步**：如果在Session期间持久态对象的状态发生了改变，Hibernate会跟踪这些变化，标记对象为“脏”对象。当Session进行flush（刷新）操作时，Hibernate会自动检测并提交所有更改至数据库，保持缓存中对象与数据库记录的一致性。

**事务边界内的对象可见性**：在同一个事务中，对持久态对象的所有操作都是基于Session缓存中的同一份数据，确保了并发操作下的数据一致性。

**OID管理**：持久态对象都有一个与数据库表中主键相对应的OID，Hibernate通过OID来管理对象在缓存中的唯一性，并在必要时执行相应的SQL语句。

**2024.3.26**

**1.事务的四特性是什么？简要说明一下四大特性：**

原子性：事务是一个原子操作单元，其对数据的修改要么全都执行，要么全都不执行。

一致性：事务必须使数据库从一个一致性状态变换到另一个一致性状态。

隔离性：事务的执行不受其他事务的干扰，事务执行的中间结果对其他事务是不可见的。

持久性：一旦事务提交，则其结果就是永久性的，即使系统崩溃也不会丢失。

**2.在不考虑隔离级别的情况下，会什么样的问题，简要说明产生的问题：**  
如果不考虑事务的隔离级别，可能会遇到以下问题：

脏读：一个事务读取了另一个尚未提交事务的修改。

不可重复读：在一个事务内，多次读取同一数据返回的结果有所不同。

幻读：在一个事务内，读取到了别的事务插入的数据行。

以上问题的出现都是因为事务之间的操作没有被适当隔离，导致数据的不一致性。

**3.事务的规范性代码如何编写，请写一个事务规范代码的模板：**

Session session = sessionFactory.openSession();

Transaction tx = null;

try {

tx = session.beginTransaction(); // 开始事务

// 操作处理

tx.commit(); // 提交事务

} catch (Exception e) {

if (tx != null) {

tx.rollback(); // 回滚事务

} finally {

session.close(); // 关闭Session

}

**4.请分别写一条查询所有的HQL语句和SQL语句：**  
**HQL：**

String hql = "User"; // User为类名

Query query = session.createQuery(hql);

List<User > results = query.list();

**SQL：**

|  |
| --- |
|  |
|  |
|  |

SELECT \* FROM user;