应用系统体系架构 — 作业8

# 学号： 522031910213 姓名： 朱涵 得分：

1. 实现MongoDB与MySQL融合的数据存储和处理方案：

思路：

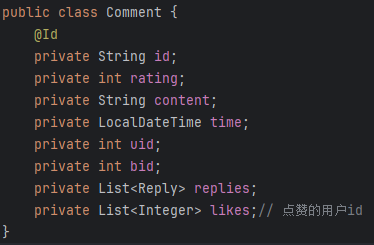
将comments、likes、replies数据存入MongoDB中，其余数据不变。

假设我的电子书店对于评论点赞回复只需要在书籍页面作展示，不需要筛选一个用户的所有点赞或回复，那么将上述数据以嵌套的形式组织在一个collection中，效率是极高的，比MySQL中三个实体表建立外键要快的多。

未来若有展示每个用户的所有reply、likes的需求，可以考虑使用图数据库存储，方便反向查找。

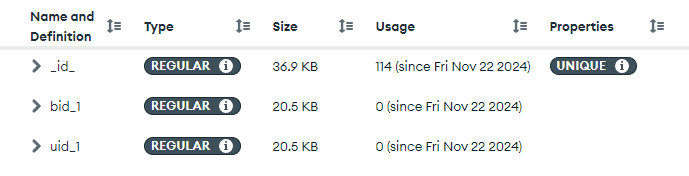
具体实现：

导入spring-boot-starter-data-mongodb依赖，将comment实体的注解改为@Document来配置表源，把Repo层改为MongodbRepository，其余逻辑实现和原本使用MySQL实现基本一致。实体属性如下：

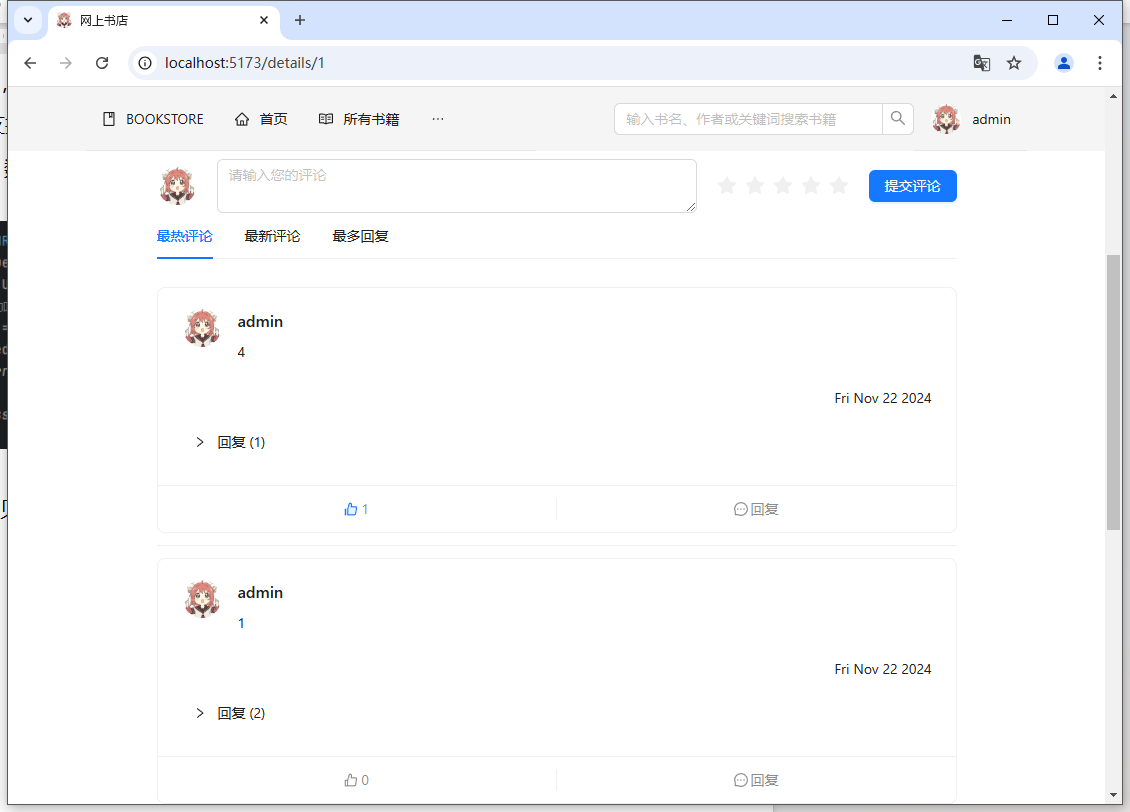


其中需注意，MongoDB默认分配id是string类型，为int会导致id只能为0；另外，MongoDB支持Partial Update，比如新增reply时，无需将整个comment取出，只需要取出replies数组进行局部更新即可省去额外开销。实现如下：



另外为了查询一本书的评论更快，我在bid字段上建立了索引方便查询。同时也建立了uid的索引方便后续实现展示每个用户的评论的功能。上面提到的展示每个用户的回复和点赞，也可以通过建立索引来加快查询，但是当数据量太大时仍会不理想。

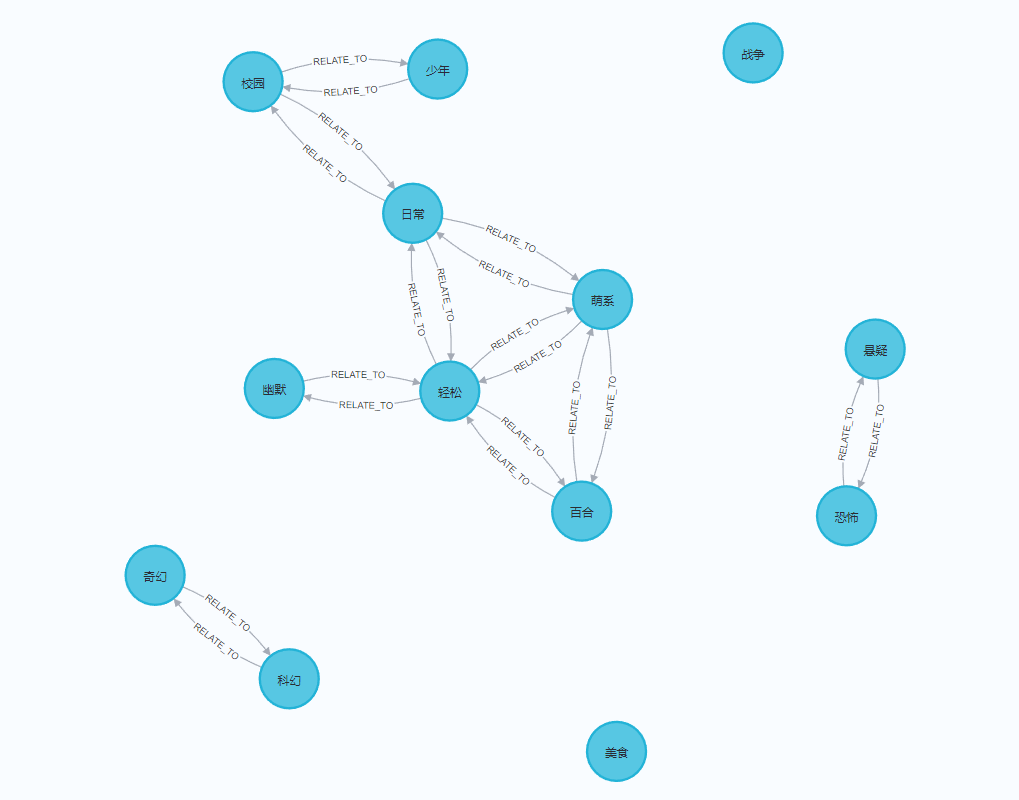
其余逻辑可见代码。前端展示如下：



1. 正确实现图书标签图在Neo4J中的构建与存储，并实现针对标签图的模糊搜索功能

答：

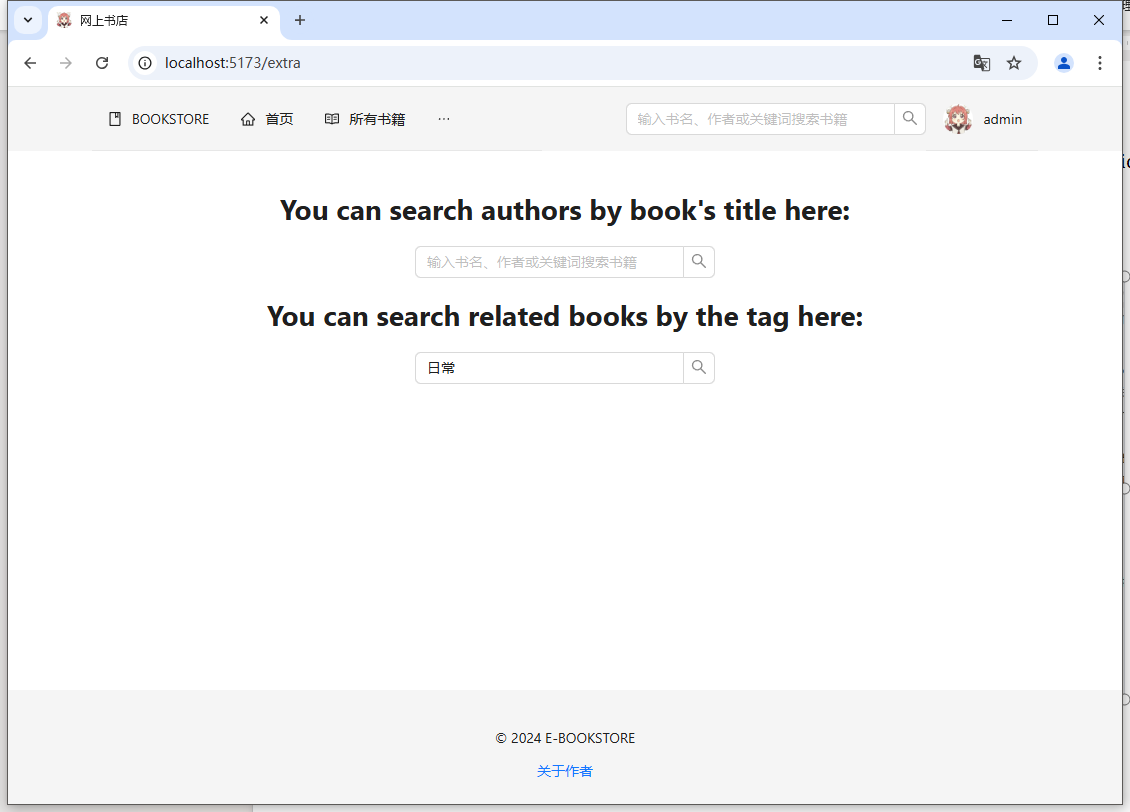
实现：

在neo4j中存储label为tag的节点，并创建label为RELATE\_TO的边，表示两个tag所包含的书籍中有较多重合（比如“日常”和“校园”），如下图

MySQL中创建book\_tags关系表记录bid和tid的关联，并新建两个接口。一个根据tid数组查询所有的book数据，一个根据输入的tid查找所有相关联的其他tag的id（也就是通过两条边可以连接到的tag）。

导入spring-neo4j的依赖，创建tag等实体、repo层，编写service层和controller层的逻辑。

前端提供一个tag筛选功能，可以选中多个tag，会展示分类下的书籍分页结果。另外提供一个模糊搜索功能，输入一个tag，会跳转到对应的搜索结果页面，展示所有相关联的tag的书籍（复用上面的tag筛选功能）。下图是前端界面



如下图是输入“日常”后点击跳转的页面，展示了所有相关tag的书籍。

