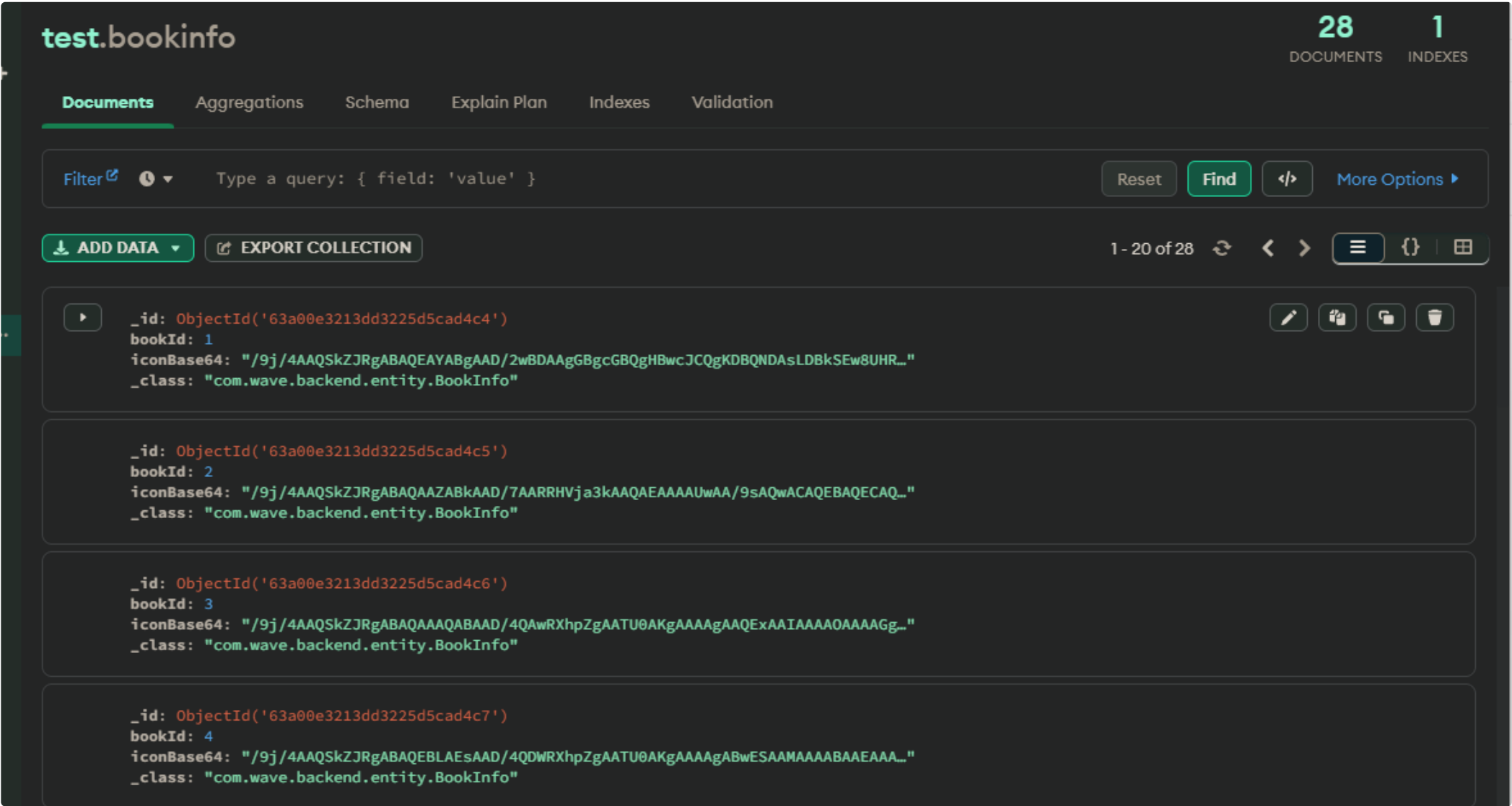


HM 11

MongoDB

我选择将图书的封面信息以 base64 的方式存储在 mongoDB 中，其他信息仍然存储在 MySQL 里面。

MongoDB 数据如下所示：



我新加了一个实体类 BookInfo，@DOCUMENT 的注解用来和 MongDB 中对应的 Document 相关联，同时为其增加了继承自 MongoRepository 的 Repository 接口类，通过这这种方式可以在不改变 Service 层代码，仅修改 Dao 层代码的情况下达到从 MySQL 和 MongoDB 两个数据源读取数据并整合返回的要求。

Java

```
@Document(collection = "bookinfo")
public class BookInfo implements Serializable {
    @Id
    private Integer bookId;
    private String iconBase64;
}

public interface BookInfoRepository extends MongoRepository<BookInfo, Integer> {
    BookInfo findBookInfoByBookId(Integer bookId);
}
```

修改后的 Dao 层代码如下，将从 MySQL 和 MongDB 中各自查找到的数据组装成一个类，这对于调用查找书籍服务的代码是不受影响的，实现了业务层和数据库层的解耦。

Java

```
public Book findById(Integer id) {
    Book book = bookMapper.selectById(id);
    BookInfo bookInfo = bookInfoRepository.findBookInfoByBookId(id);
    if(bookInfo != null){
        book.setCover("data:image/jpeg;base64," + bookInfo.getIconBase64()); // jpeg格式压缩大小
        log.info("get image base64 from mongoDB");
    } else {
```

查找效果如图：

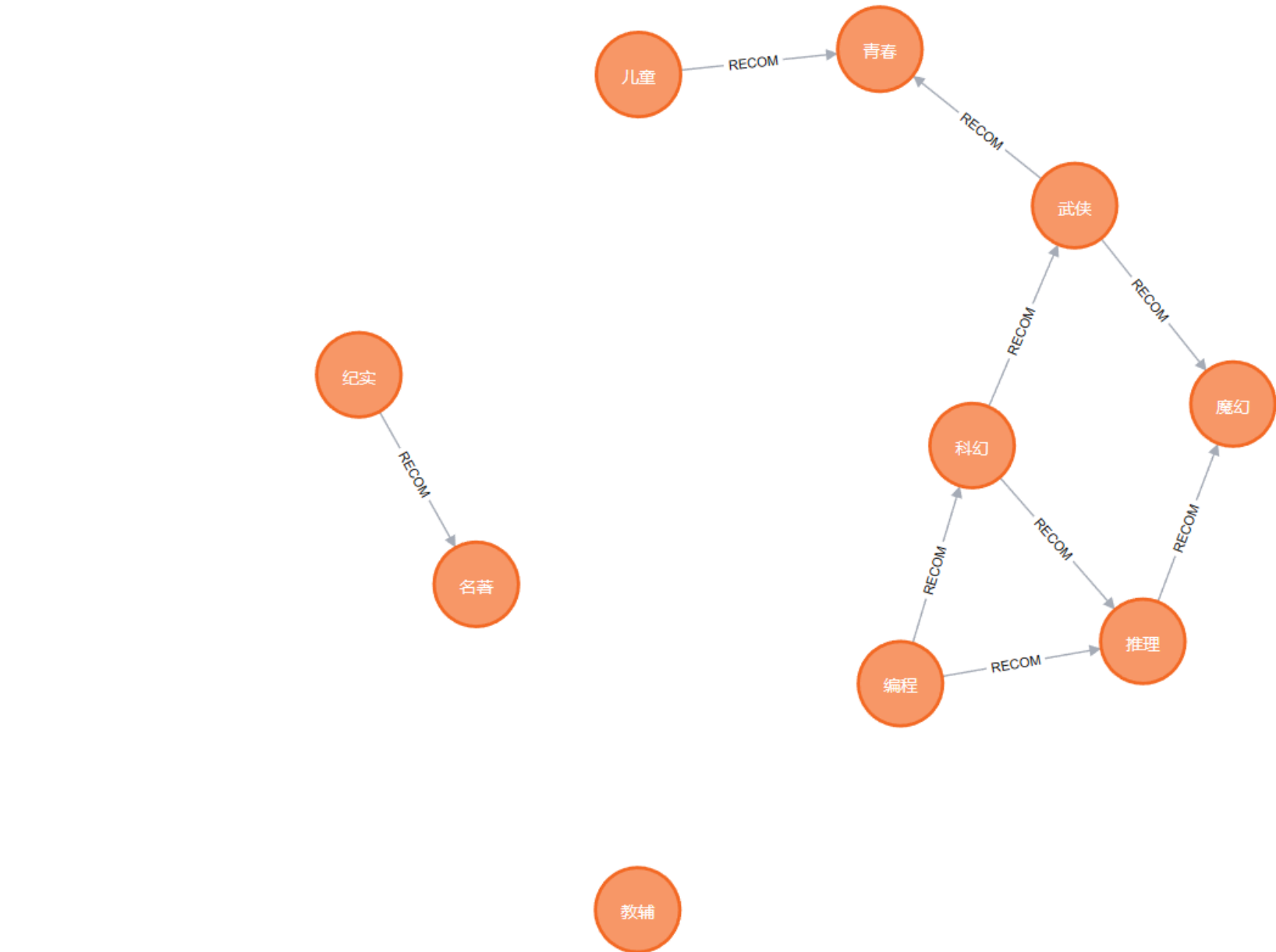
```

Initializing Servlet DispatcherServlet
2022-12-19 14:27:26.070 INFO 32396 --- [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : 
Completed initialization in 2 ms
Book: 1 is not in Redis
Searching Book: 1 in DB
2022-12-19 14:27:27.896 INFO 32396 --- [nio-8080-exec-2] org.mongodb.driver.connection : 
Opened connection [connectionId{localValue:3, serverValue:279}] to 127.0.0.1:27017
2022-12-19 14:27:27.934 INFO 32396 --- [nio-8080-exec-2] c.wave.backend.dao.daoImpl.BookDaoImpl : 
get image base64 from mongoDB

```

我为数据库中的每一本书都添加了一个或多个标签（Label），每个标签都有自己的推荐标签，当用户查找当前标签的书本时，拥有该标签的推荐标签的书本也会被同时推荐。

将书本标签的上述关系构建成为一张连通图，保存在 Neo4j 中，如图所示（并非所有标签都有推荐）



标签类的编写如下：

Java

```
@Node
public class BookLabel {
    @Id @GeneratedValue
    private Long id;
    private String labelName;
    private Integer labelId;
    public BookLabel(){
    }
    private BookLabel(String labelName){
        this.labelName = labelName;
    }
    @Relationship(type = "RECOM")
    public Set<BookLabel> recomb;
    public void addRecomLabel(BookLabel bookLabel){
        if(recomb == null){
            recomb = new HashSet<>();
        }
        recomb.add(bookLabel);
    }
}
```

根据标签名进行二重关系查找的查询代代码下（一重关系直接读取 Label 的成员即可，二者取并集即为结果）

Java

```
@Query("Match (n:BookLabel {labelName:$labelName})-[:RECOM]->()-[:RECOM]->(m:BookLabel) RETURN m")
List<BookLabel> getLabelsInTwoSteps(String labelName);
```

由于设置了二重查找关系，因此当用户查找某个标签的时候，该标签的推荐标签以及推荐标签的推荐标签均会被推荐给用户，例如当用户查找 编程 标签的时候， 科幻 ， 推理 ， 武侠 ， 魔幻 都会被推荐给用户。（暂且不考虑这样的推荐合不合理），效果如图：

欢迎来到EBook

退出登录

评论

根据编程标签为您推荐相关书籍


编程

搜索


关闭

标签推荐


购物车




魔兽世界编年史套书...
作者:克里斯 梅森
价格: 449.2元




三体: 全三册
作者:刘慈欣
价格: 50.2元




纳尼亚传奇
作者:刘易斯
价格: 86.2元




魔力的胎动
作者:东野圭吾
价格: 35.9元




天龙八部(全五册)
作者:金庸
价格: 102.3元



笑傲江湖(全四册)
作者:金庸
价格: 80.1元



楚留香传奇(全三册)
作者:古龙
价格: 224.5元



哈利波特与魔法石
作者:J.K.罗琳
价格: 30.2元