



链滴

关于文档型数据库的一点理解

作者: [gfafei](#)

原文链接: <https://ld246.com/article/1495070917008>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

最近用Node+MongoDB写了一个论坛，期间遇到一些麻烦，也做了一些思考。在此记录一下。

问题

首先表结构是这样的：

```
User: {
  username: String,
  pass: String
}
```

```
Topic: {
  creator: ObjectId,
  tab: String,
  title: String,
  content: String
}
```

```
Reply: {
  creator: ObjectId,
  replyId: ObjectId,
  topicId: ObjectId,
  content: String
}
```

为了便于说明，这里做了一些简化

点开一个帖子，里面应该显示帖子的创建者以及所有回复。这时后台要做的事就很麻烦：

- 查出帖子
- 查出帖子的创建者姓名
- 查帖子的所有回复
- 查回复的创建者姓名

这里的逻辑写起来非常别扭。为了前台使用的方便，需要把`topic.creator`替换成`user`文档。而根据mongoose的设计，查出的数据和数据库里保持一致，如果改变就要save，否则不能读取改变后的数据。从这里都要调用`toObject`方法，才能将`topic.creator`替换成`user`文档，这样的话mongoose的封装就有意义了。

再者，前台发出一个请求，后台就要和数据库交流这么多次，这是不合理的。这种拼接对象的逻辑也是control层该干的事。

思考

这个论坛的代码结构很大程度上参考了nodeclub这个项目，而这个项目是很久之前的，以现在的角度看，它的设计未必没有问题。[MongoDB 进阶模式设计](#)这篇文章给了我很大的启示。

mongoDB是文档型nosql数据库，选择了mongoDB就应该改变关系型数据的思维。数据不是一个有关系的实体，而是一些互相之间关联较少，内部结构复杂的文档。在MongoDB中,应该优先考虑内，比如话题的创建者可以作为一个文档内嵌在话题中。

```
Topic: {
  creator: {
    _id: ObjectId,
    username: String
  }
  ...
}
```

这样做就能一次读取需要的信息，大大提高了读取性能，造成的后果是，改变用户名的代价变高了。改变用户名的机会是比较少的，或者干脆不允许修改用户名问题也不大。所以这样做是比较划算的。

需要注意的是，mongoDB对单个文档的大小是有限制的：16M。有的时候为了保持可扩展性，就不使用内嵌。比如将回复看作话题的属性，那回复的内容和数量就会受到这个16M的限制，不能无限扩展了。

另外，帖子的回复数量和最后回复时间这样的信息可以从回复表中获取，但也可以看作帖子的一个属性，内嵌在Topic表中。这就有一个数据冗余的问题：话题上记录的回复数量和回复表中该话题的回复量不一致怎么办。有两种办法：一是每次添加或删除回复都去更新topic表，二是认为帖子的回复数一定时间内不会改变太多。比如将这个时间定为三个小时。

```
Topic: {
  replyCount: {
    timeStamp: Date,
    value: number
  }
  ...
}
```

读取回复数时，如果数据是三个小时之内的就认为数据是大致准确的，否则从回复表中读取数量。这其实相当于一个缓存。