在研究了一段时间后，发现ES不适合做为数据存储使用，理由如下：

1. ****mapping不可改，不能改index属性。****ES中以定义的mapping不能修改名字和属性，无法修改名字勉强还能接受，但无法修改属性。官方文档中介绍了几种修改mapping的方法。一个是新建一个字段，程序中所有地方修改名字，这对于复杂的项目容易出错，而且无法保留原来的数据；另一个是利用alias创建一个新的索引，但是所有数据需要重新导入，这需要很长时间，操作性不强。
2. ****无法多对多。****ES中提供3种关联关系：Field collapsing（严格来说不是关联），Nested object，Parent-child。前两种都是直接将一个mapping声明在另一个mapping中，第三种关联是在创建子文档是指明他的父文档，但是一个子文档只能有一个父文档，因此也不能实现多对多的关联。其实如果理解了ES的目的是提升检索效率，就不难理解为什么没有多对多关联了，在关系数据库里这就是个效率瓶颈。
3. ****没有用户验证和权限控制。****ES本身的访问权限可以通过nginx进行控制，但是同一个ES中不同索引间目前是没有权限控制的。
4. ****从ES设计的初衷看，为了检索，为了统计。****这个从字段的store属性中可以看出来，查看ES手册（https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-store.html）可以发现，默认情况下字段的原始值是不会被保存的，这跟数据存储是南辕北辙了。
5. ****项目开始时不好确定shards数量。****少了以后扩展不方便，多了一开始影响性能。这个可以通过将type命名为doctype-yyyymmdd来解决，每天都生成新的一个或多个shard，但是注意在搜索时需要在doctype-\*中搜索。
6. ****ES非常适合特定的需求，但不适合用于数据存储。****ES索引速度快，扩展方便，性能优异，但在功能上不适合作为数据库使用。数据存储的目的是为了以后能方便的使用，不仅是针对当前的需求，也要为未来可能出现的需求做准备。由于ES有以上几点问题，无法适应需求变化。

ES适合的场景

1. 检索。ES本身作为一个搜索引擎，用来处理检索的任务再合适不过。你可以在线上项目中直接将内容写入ES以提供检索服务，也可以把以往的数据导入ES以处理特定的需求。关于ES和Solr的比较以后有时间的话会写一篇
2. 统计。ES的统计也是基于检索功能的，聚合功能使得统计结果处理起来非常方便。如果你只需要统计而不用检索，可能有其他工具更适合你，比如Spark SQL。