

基于MongoDB文档模型的教学资源数据的建模研究

黄承明

(湖南财经工业职业技术学院, 湖南 衡阳 421002)

✉vip71800@163.com



摘 要: 教学资源数据几乎全部都是半结构化数据和非结构化数据, 传统应用中采用RDBMS和文件系统进行存储存在诸多的弊端。文章分析了教学资源数据组成结构与应用特性, 以及MongoDB数据库的存储机制, 提出了基于MongoDB + GridFS的存储方案, 并结合实际案例总结出了教学资源数据逻辑建模方法和建模要点。

关键词: 半结构化; 非结构化; MongoDB; 文档模型; 数据建模

中图分类号: TP31 **文献标识码:** A

Modeling of Teaching Resource Data Based on MongoDB Document Model

HUANG Chengming

(Hunan Finance Economics & Industry Polytechnic, Hengyang 421002, China)

✉vip71800@163.com

Abstract: Almost all of the teaching resource data are semi-structured or unstructured. In traditional applications, there are many disadvantages in using RDBMS and file system for storage. This paper analyzes the data structure and application characteristics of teaching resources and the storage mechanism of MongoDB database, and puts forward a storage scheme based on MongoDB + GridFS. Combined with practical cases, the paper summarizes the logical modeling method and important aspects of teaching resource data.

Keywords: semi-structured; unstructured; MongoDB; document model; data modeling

1 引言(Introduction)

教育信息化包含了一系列的目标、价值观念、技术规范以及应用手段等内容, 其核心内容是教学信息化, 需要各种各样的信息化教学应用系统作为支撑, 数字化的教学资源数据是所有这些应用系统的基础。

《教育信息化2.0行动计划》明确指出了数字资源建设的重要性, 要科学、合理、高效地进行数字化教学资源的建设、完善、分发和共享^[1]。对于结构和类型日益复杂的数据库, 传统关系型数据库暴露出很多问题, 如磁盘IO瓶颈、模式固定难以横向扩展等, 带来较高的维护成本和较低的可用性^[2]。NoSQL数据库的出现为这一困境提供了新的解决问题的途径。

2 教学资源数据解析(Data analysis of teaching resources)

广义的教学资源是指支持教学活动的各种资源, 包括人、财、物、信息等。可分为人类资源(教师、学生、学习小组等)和非人类资源(各种媒体资源, 辅助工具及设施, 社会教

育性机构, 如: 试听中心、图书馆、博物馆、少年宫等)。本文中的教学资源仅指狭义上的教学资源, 即教学资料、教学环境及教学支撑系统。

数字化教学资源是指经过数字化处理后, 以数字信息的形式存在于计算机存储介质中、通过计算机及网络进行传播、运行的教学资源, 包括: (1)基本形态资源, 即素材资源, 如文本、数据、图形图像、音频、视频、动画等; (2)复合型态资源, 即应用资源, 如题库、课件、案例库、知识库、课程库等^[3]; (3)应用系统, 如图文系统、教学应用系统、仿真系统等。

教学资源数据是指数字化教学资源中的素材资源和应用资源在计算机中的存储形态。这些教学资源数据按照一定组织结构存储在指定的位置, 所形成的教学资源集合称为数字化教学资源库。

要实现数字化教学资源在计算机中的存储、识别和管理, 除了要保存数字化教学资源本身数据外, 还需要大量的辅助性的数据来描述其各项属性, 如名称、类型、格式、关

系、位置、创建时间、版本等，其中资源本身数据称为本体数据，描述资源本体数据的数据称为元数据。

根据数据的结构化程度，可以将教学资源数据分为两大类：

非结构化数据，几乎所有的素材资源本体数据都是非结构化数据，如图片、图形、动画、音频、视频、仿真模型等，这些数据都没有固定的结构，也就没有数据之间的关系存在，通常以二进制格式进行整体存储^[4]。

半结构化数据，部分素材资源本体数据(如，词条)、所有的应用资源本体数据、教学资源元数据，这些数据没有严格的格式，数据之间关系松散，具有下述典型特点^[5]：

- ① 自描述性。先有数据，再有结构；
- ② 不精确性。随时间和场景变化；
- ③ 不规则性；
- ④ 非强制性；
- ⑤ 模式复杂性，也称模式自由或无模式。

目前信息化教学应用中的存储方案为：(1)半结构化数据进行规范化，存储在关系型数据库中；(2)非结构化数据以文件的形式存储在计算机存储系统中，通过路径名和文件名进行访问。

教学资源数据在应用上具有以下典型特征：(1)写少读多，应用成分布式；(2)种类复杂，功能迭代快，扩展性要求高；(3)文件数量极多，小文件比重大。

由于教学资源数据的数据特点和应用特征，传统的存储方案存在较为严重的缺点：(1)扩展难度大，如增加资源数据属性，资源数据属性的不一致性、嵌套属性等导致数据结构冗余度高；(2)当文件的数量庞大，分类复杂及多重分类时，文件系统管理难度增大，性能随文件数量增大而降低。

针对传统存储方案的不足，建议采用基于NoSQL数据库的存储方案：MongoDB + GridFS文档型数据库存储方案。

3 文档模型存储机制(Storage mechanism of document model)

MongoDB是一个基于分布式文件存储的NoSQL文档模型数据库，因其高效的性能和丰富的使用功能在生产中得到非常广泛的应用。它的体系结构及同关系型数据库对照概念如表1所示。

表1 MongoDB与RDBMS的概念对应

Tab.1 Concept correspondence between MongoDB and RDBMS

RDBMS概念	MongoDB概念	说明
database	database	数据库
table	collection	表与集合等价
row	document	记录行等价文档
column	field	数据列/字段
index	index	索引
table joins		MongoDB不支持表连接
primary key	primary key	MongoDB默认_id字段为主键

Document是MongoDB的核心对象，是数据的基本组织单元，类似关系型数据库表的行，但是比行要复杂，它是使用BSON这种结构来存储数据和进行数据交换。BSON是一

种类JSON的二进制形式的存储格式，简称Binary JSON，它比JSON支持更多的数据类型，如二进制数据^[6]。典型的Document对象如图1所示。

```
{
  "_id": ObjectId("5d59469688018c76d32cf20"),
  "surname": "张",
  "pname": "小五",
  "gender": "男",
  "isTeacher": false,
  "isStudent": true,
  "dateOfBirth": ISODate("2001-01-01T00:00:00.000+0800"),
  "phone": {
    "mobile": "13800212290",
    "work": "86-21-83002109"
  },
  "hobby": ["阅读", "登山", "高尔夫"]
}
```

图1 MongoDB的Document对象

Fig.1 Document object of MongoDB

Field是MongoDB数据库的基本数据单元，采用key:value键值对来表示数据，其中key用字符串表示，value则为MongoDB所支持的任何数据类型，如：Null、Boolean、String、Symbol、Object Id、Date、Regular Expression、Code、Array、Embedded Document等^[6]。

BSON格式的Document对象是一种无模式(schema-less)的存储形式，能够动态地添加平行或嵌套的数据项，灵活性很高，具有三个典型的特点：轻量性、可遍历性、高效性。非常适合存储教学资源数据中的半结构化数据。

在MongoDB中BSON格式的单个Document对象(含嵌入的子Document)受到16MB大小的限制。所以，当key:value中的value值尺寸过大时，就要保存到GridFS中。

GridFS是一种将大型文件存储在MongoDB的文件规范^[7]。GridFS将文件分成多个块存储，每块作为一个单独的文档，默认大小为255kB，最后一个块为剩余大小。对于存储海量小文件在性能上仍然能够优于文件系统^[8]。

一个文件在GridFS的存储结构如图2和图3所示，使用两个集合files和chunks存储，为一对多关系。

Key	Value	Type
question_bank		
media_file.chunks		
media_file.files		
Views (0)		
GridFS Buckets (1)		
media_file		
System (0)		

图2 文件在GridFS中存储示意图(files集合)

Fig.2 Files storing footprints in GridFS(files collection)

Key	Value	Type
question_bank		
media_file.chunks		
media_file.files		
Views (0)		
GridFS Buckets (1)		
media_file		
System (0)		

图3 文件在GridFS中存储示意图(chunks集合)

Fig.3 Files storing footprints in GridFS(chunks collection)

采用GridFS存储教学资源数据中的图片、音频、视频等非结构化的数据具有几个优势：

- (1)可以简化需求，使用GridFS后就可以不需要另外使用独立文件存储架构，从而实现全部教学资源数据的一体化存取。
- (2)GridFS能够直接利用MongoDB已建立的复制/分片机

制, 非常容易实现分布式的扩展、文件分发和故障恢复。

(3) 无须关心放置新文件的位置, 没有因为文件数量增大、目录加深时严重影响系统读写性能的问题。

(4) GridFS不产生磁盘碎片。

4 建模方法(Modeling method)

在技术上, MongoDB是一种无模式(或者自由模型)的文档型数据库, 但是, 在实现应用系统的功能时, 仍然需要采用规范约定、技术手段来实现数据的业务模式, 主要是表现数据之间的一定的层次和关系。因此, 教学资源数据的建模在数据库逻辑设计阶段显得尤其重要, 目的在于提高系统性、优化业务功能、方便团队沟通和提高生产效率。

在RDBMS中, 为了使数据结构清晰、关系明确、逻辑严格, 需要对数据对象拆分成原子数据项, 按照相关范式进行规范化处理, 需要很多的主从结构数据表来存储, 建立主外键来描述关系, 查询时还需要进行连表^[9]。

由于MongoDB的文档模式支持数组和内嵌文档, 可以直接用一个Document来描述一个数据对象, 而不需要分表/集合存储。在同一个集合中, Document的数据项并没有强制要求完全一样, 完全可以将不同类型的数据放在一个集合中, 例如, 我们可以将衣服、手机放在product集合中, 尽管衣服和手机两类产品具有部分不同的属性。

此外, 在MongoDB数据库中, 每个Document默认包含一个名为“_id”的ObjectId类型属性, 用来唯一标识一个文档对象, 类似于RDBMS中的主键, 为文档之间的对象引用提供了技术支持。

所以, 在基于MongoDB数据库的教学资源数据的业务建模中, 可以采用反规范化处理和规范化处理是两种方法来进行建模。

4.1 内嵌模式的反规范化处理

内嵌模式是指Document中的一个key:value中的value值也是一个Document对象, 如图4所示。

```
//User Document
{
  "_id": ObjectId("5d04e8a076d44e5e7cc7d97d"),
  "fullName": "小白",
  "age": 21.0,
  "home": {
    "phone": "010-12345678",
    "street": "西三环100号",
    "city": "北京"
  },
  "work": {
    "phone": "022-12345678",
    "city": "天津"
  },
  "groups": [
    { "gname": "开发3组", "position": "主管" },
    { "gname": "产品1组", "position": "组员" }
  ]
}
```

图4 典型的Document对象

Fig.4 Typical document object

这种结构在RDBMS中是反规范化的^[10], 但是这种反规范化处理的方法能够直观地表示一对一的关系, 如图4中的home、work子对象, 好处是一次查询即可获得完整的对象, 不需要再额外查询其他集合的文档对象。通过内嵌对象数组可以实现一对多的关系, 多级的内嵌对象还可以方便地实现树形/层次结构, 如图4中groups数组对象。其缺点是当嵌入对象被重用时, 要在每个文档中都存储它的一个独立的副本, 这会减慢插入的速度, 还占用额外的磁盘空间。

4.2 引用模式的规范化处理

通过MongoDB的引用模式可实现类似RDBMS中的外键关系, 以解决重用子对象的多副本问题, 主要的目的是减少冗余和优化依赖关系。

以素材资源存储为例, 涉及两个数据对象: 素材和人员, 由于素材资源的本体数据(文件)是存储在GridFS中, 不能内嵌, 必须引用, 所以就变成素材属性、素材文件和人员三者之间的关系, 而人员信息因为需要重用, 最好引用, 参考图5。

```
//Person Document
{
  "_id": ObjectId("5d5942dc88018c7f6d32cf20"),
  "surname": "张",
  "pname": "小五",
  "gender": "男",
  "dateOfBirth": ISODate("2001-01-01"),
}
//File Document
{
  "_id": ObjectId("5d5942dc88018c7f6d32cf0f"),
  "fileName": "计算机字符编码.pdf",
  "chunkSize": NumberLong(261120),
  "uploadDate": "2019-08-18T12:21:48.102",
  "length": NumberLong(607604),
  "contentType": "application/pdf",
  "md5": "ab21e359089704cadcd694e5370396c1"
}
//Media Document
{
  "_id": ObjectId("5d5942dc88018c7f6d32cf2d"),
  "fileName": "计算机字符编码.pdf",
  "categories": ["C00009", "C00221", "0008"],
  "tag": ["字符编码", "UTF-8", "GBK", "ANSI"],
  "add_man": "5d5942dc88018c7f6d32cf20", //人员引用
  "file_id": "5d5942dc88018c7f6d32cf0f" //文件引用
}
```

图5 引用Document对象

Fig.5 Reference document object

在Media Document(素材属性)中的add_man、file_id就是通过对File Document(素材文件)和Person Document(人员)的“_id”进行引用, 从而形成了典型的一对一或一对多的对应关系。

4.3 多对多关系的实现

用过引用模式实现了一对多的对应关系, 则多对多的关系也容易实现, 最典型的例子是人员与分组之间的对应关系, 参考图6。

```
//Person Document
{
  "_id": ObjectId("4e54e09f48dc3722c0094a43"),
  "surname": "张",
  "pname": "小五",
  "gender": "男",
  "groups": [
    ObjectId("4e54e09f48dc3722c0094a42"),
    ObjectId("4e54e09f48dc3722c0094a41")
  ]
}
//User Document
{
  "_id": ObjectId("4e54e09f48dc3722c0094a43"),
  "surname": "张",
  "pname": "小五",
  "gender": "男",
  "groups": [
    ObjectId("4e54e09f48dc3722c0094a42")
  ]
}
//Group Document
{
  "_id": ObjectId("4e54e09f48dc3722c0094a42"),
  "groupName": "数据资源项目组",
  "persons": [
    ObjectId("4e54e09f48dc3722c0094a43"),
    ObjectId("4e54e09f48dc3722c0094a40")
  ]
}
{
  "_id": ObjectId("4e54e09f48dc3722c0094a41"),
  "groupName": "数据资源项目组",
  "persons": [
    ObjectId("4e54e09f48dc3722c0094a43")
  ]
}
```

图6 多对多关系

Fig.6 Many to many relationship

MongoDB中的这种引用模式所形成的一对多、多对多关系同RDBMS一样, 同样是节约了空间但牺牲了性能。然而, 对用需要重用的引用对象来说, 只有如此才能确保数据的一致性。

4.4 典型案例

(1) 素材资源

素材资源是最基本的教学资源, 也是数量最多的一种资源, 典型的数据模型为“资源属性数据(元数据)资源文件数据(本体数据)”模式, 其文档模型如图7和图8所示。

```
{
  "_id": ObjectId("5d594c88018c7f6d32cf2d"),
  "fileName": "计算机字符编码.pdf",
  "format": "pdf",
  "categories": ["C02069", "C03221", "808"],
  "tag": ["字符编码", "UTF-8", "GBK", "ANSI"],
  "add_date": "ISODate('2019-08-18T00:00:00.000+0800')",
  "upload_date": "ISODate('2019-08-18T00:00:00.000+0800')",
  "length": "NumberLong(261120)",
  "contentType": "application/pdf",
  "md5": "5d594c88018c7f6d32cf2d"
}
```

图7 素材资源的属性数据

Fig.7 Attribute data of material resources

```
{
  "_id": ObjectId("5d5942dc88018c7f6d32cf0f"),
  "fileName": "计算机字符编码.pdf",
  "aliases": null,
  "chunkSize": NumberLong(261120),
  "uploadDate": ISODate('2019-08-18T12:21:48.102+0800'),
  "length": NumberLong(687604),
  "contentType": "application/pdf",
  "md5": "ab21e359809794cad694e5379396c1"
}
```

图8 素材资源的文件数据

Fig.8 File data of material resources

在属性数据中采用了引用模式对文件数据进行了关联。

(2)题库资源

题库资源是典型的应用资源,是教学应用中使用频率最高的资源之一。题库资源在RDBMS中的逻辑模型比较复杂。其关系模型如图9所示。

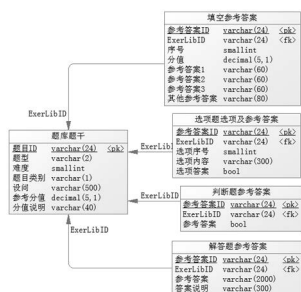


图9 题库的关系模型

Fig.9 Relation model of item library

但是,在MongoDB中采用聚合的方式,可以存放到一个集合之中,其数据模型如图10所示。

```
{
  "_id": ObjectId("5d5942dc88018c7f6d32cf0f"),
  "exerType": "单选题", // 单选题、多选题、判断题、填空题、简答题
  "difficulty": "2", // 难度系数
  "score": "1", // 分值
  "category": "计算机组成原理", // 类别
  "author": "计算机组成原理", // 作者
  "name": "计算机组成原理", // 名称
  "fill_in_answer": [ // 填空题答案
    { "id": "1", "answer": "计算机组成原理", "score": 1 },
    { "id": "2", "answer": "计算机组成原理", "score": 1 }
  ],
  "choice_answer": [ // 选择题答案 (单选、多选)
    { "option": "A", "isAnswer": true },
    { "option": "B", "isAnswer": false },
    { "option": "C", "isAnswer": false },
    { "option": "D", "isAnswer": false }
  ],
  "true_false_answer": [ // 判断题答案
    { "id": "1", "answer": "计算机组成原理", "score": 1 },
    { "id": "2", "answer": "计算机组成原理", "score": 1 }
  ],
  "item_analysis": [ // 题目分析
    { "id": "1", "analysis": "计算机组成原理", "score": 1 },
    { "id": "2", "analysis": "计算机组成原理", "score": 1 }
  ]
}
```

图10 题库的文档模型

Fig.10 Document model of item library

需要注意的是,本文档模型是一个聚合模型,其中的四个参考答案(fill_in_answer、choice_answer、brie_answer、true_false_answer)可以根据exerType的不同取值进行选用。同时可以根据exerType进行分片部署,以提高检索效率。

5 建模要点(Key points of modeling)

教学资源数据建模是构建教学应用系统的关键环节,直接影响到系统的性能和可用性。所以,在进行数据建模特别要注意以下几点。

(1)数据建模必须以满足业务需求为首要目标,切不可因追求技术的完美而脱离业务规则。

(2)在建模过程中,要灵活使用各种设计模式,切不可生搬硬套。

(3)在满足业务需求的前提下,一切以性能优先,尽量全

面考虑,优化读写性能。

同时,在设计过程中遵循基本的设计原则,首先,模型设计是为应用系统服务,而不是为了存储空间优化;其次,要为实现系统最佳性能设计,可以通过合理的索引、数据分片来提高系统的存取性能。

此外,特别需要注意的是引用和内嵌的选用问题,需要做到优先选用与平衡原则的使用。

(1)在选用内嵌模式和引用模式时,优先考虑内嵌,系统性能优先存储空间。

(2)内嵌模式适合一对一、一对多的小对象嵌入,其局限性主要有两个:16MB体积的限制,大数组的低性能约束。

(3)引用模式适合一对多、多对多中的对重用对象的关联,多次查询、写入数据时不能实现跨集合的事务性是该模式的主要缺陷^[11]。

6 结论(Conclusion)

综上所述,采用MongoDB+GridFS存储方案代替RDBMS+文件系统存储方案在技术上是可行的,在表达半结构化及非结构化数据的业务逻辑上具有天然的优势,在功能上能够提供更好的一致性、灵活性和扩展性,同时具有较好的系统性能。MongoDB数据库与GridFS文件规范相对RDBMS和文件系统来说属于非常年轻的存储技术,有很大的优化空间,而基于MongoDB+GridFS存储方案的信息化教学应用系统的实际性能、鲁棒性有待进一步的验证。

参考文献(References)

- [1] 吴旻瑜,武晓菲.教育信息化2.0的时代逻辑——《教育信息化2.0行动计划》解读之一[J].远程教育杂志,2018(4):4-10.
- [2] 宗平,李雷.PostgreSQL与MongoDB处理非结构化数据性能比较[J].计算机工程与应用,2017,53(7):104-108.
- [3] 黄承明.基于MongoDB的数字化教学资源存取方案研究[J].中国电化教育,2018(3):54-57.
- [4] 张孝,周宁南.非结构化数据存储管理研究[J].科研信息化技术与应用,2013,4(1):30-40.
- [5] 王静,孟小峰.半结构化数据的模式研究综述[J].计算机科学,2001,28(2):6-10.
- [6] 霍多罗夫.MongoDB权威指南(第2版)[M].北京:人民邮电出版社,2014:7-10.
- [7] 邱新忠.基于MongoDB GridFS的地图瓦片数据存储研究[J].地理空间信息,2016,14(2):50-52.
- [8] ITKeyword.MongoDB的GridFS与文件系统在小文件存储的读取性能对[EB/OL].http://www.itkeyword.com/doc/2118899009621598745/mongodb,2015-05-14.
- [9] 丁智斌,石浩磊.关系数据库设计与规范化[J].计算机与数字工程,2005(02):114-116.
- [10] 黄艾卿.数据库设计中的反规范化技术[J].百色学院学报,2007,20(6):88-90.
- [11] 宣超.基于MongoDB的事务机制研究与实现[D].四川:电子科技大学,2018.

作者简介:

黄承明(1968-),男,本科,副教授.研究领域:应用软件系统分析与设计,教育信息化。