山东大学 软件 学院

非关系型数据库 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：201600301079 | 姓名： 崔玉峰 | | 班级： 2016级软件4班 |
| 实验题目：Nosql实验总结 | | | |
| 实验学时：2 | | 实验日期： 2018/12/14 | |
| 实验目的：  Nosql实验总结报告 | | | |
| 硬件环境：  PC  CPU : intel-i5  RAM: 8G | | | |
| 软件环境：    数据库： MongoDB  高级程序设计语言：java  IDE : Navicat for MongoDB ,IntelliJ IDEA | | | |
| 1. **MongoDB数据库设计：**   起初对于非关系型数据库的设计不是很了解，当时的想法很简单就是认为非关系型数据库中不应该有关系表，所以就将关系表中的数据合并到实体表中去，并且MongoDB中允许出现数组结构，可以用来表示一对多的关系。  student 集合：  SID NAME SEX AGE BIRTHDAY **COURSES**(选课数组)  course 集合：  CID NAME FCID CREDIT **STUDENTS**(选课学生数组) **TEACHER**(任课老师)  teacher 集合：  TIDNAME SEX AGE BIRTHDAY **COURSES**(教课数组)  这样设计也会出现一些问题，比如要查询一个学生所有选课的名字的额时候还是需要跨表查询的会消耗大量的时间，但是如果将课程内容全部不全就大大加重了数据库的冗余，非常的得不偿失，因此还是坚持这个数据库的设计。   1. **基本增查改功能**   学习一个数据库最基础的功能就是增删改查的功能，MongoDB中的增删改查的功能的查询结构也是json格式的，通过变换某几个字段的关键字可以实现比较复杂的功能。  **插入：**  因为MongoDB文档型非关系数据库，储存json结构的文档数据，所以可以很方便的进行插入操作，甚至一个集合中json文档的格式都可以不统一，但是为了可用性，还是尽量使一个集合中的文档格式基本相同。  **查询：**  因为JSON结构可以理解为一种对象结构，所以可以访问一个json文档中的每个字段。以及其中嵌合体的每个字段，在配合逻辑运算符和算数运算符就可以实现比较复杂的查询，但是想要执行复杂的查询还是需要高级程序设计语言或者聚合操作。  **更新：**  MongoDB中的更新还是很强大的，MongoDB中给出了很多的更新字段，可以实现很复杂的功能，不光能更新某个字段数据，嵌合体字段数据；**甚至还能更新数组字段，为数组增加新的项($addToSet)，删除某一项（$pull）。**通过这个功能可以很方便的更新选课数组中元素的值，     1. **聚合的应用**     这个功能应该是MongoDB中使我感到最惊艳功能了，他与关系型数据中的聚合类似，只是它更像是一种处理数据的管道，它允许你多次进行group操作，多次sum操作，并且附加排序，筛选结果的功能。通过这些功能就能完成一个非常复杂的数据分析问题，在实验七中的应用非常好，可以大大加快检索数据所使用的时间。  因为在做刚开始第七次实验时，并不知道mongoDB的聚合操作就全部用java实现了一遍，后来在学习了聚合之后由通过聚合实现了一边，两次的运行时间还是有明显差距的，聚合操作往往比Java实现快5倍,这个速度的提升还是非常大的，所以在真实使用MongoDB数据库时可以多运用聚合操作处理问题。  具体时间对比表：   |  |  |  | | --- | --- | --- | |  | **聚合操作时间（ms）** | **Java遍历时间（ms）** | | **第二题** | **52** | **295** | | **第三题** | **62** | **319** | | **第八题** | **47** | **202** | | **第九题** | **32** | **219** |       这是我感觉实验中比较有亮点的一个地方，通过聚合操作的使用，可以大大加快查询，分析数据的速度，还是非常有特色的。尤其是在求解TOP10查询任务时，可以非常的快捷方便。  具体每个聚合语句的实现在实验报告中已经详细给出了，此处只给出速度对比。   1. **数据可视化** 2. 对于所有的实验的结果均通过界面展示，更加的美观直接，并且提供了操作接口可以很轻松的实现对数据库的操作。     **A13UC20[]BE)~6EI$%$2PSB**    虽然界面并不是本次课的硬性要求，但是作为大三的学生最后交出一个没有用户界面的命令行程序，总感觉有些不妥，于是就通过一些简单的UI库实现了一个较为美观简洁的界面     1. 数据分布的分析可视化：   通过散点图，直方图和饼图的形式对对数据的分布特点进行了可视化处理，但是因为数据的分布实在太假，很多数据都是相同的（很多学生选课相同且分数全部是145）所以继续进行模型的训练分析的意义不大。  但是通过数据可视化操作还是可以看到数据的一些分布特点的，比如已下图为例：每个选课人数分布直方图，可以看到大部分的选课人数在100~200之间，个别超过了500，分布不是很均匀，说明数据的真实性有待确定。    **然后其他的数据分布图：**   1. 课程选课人数分布图      1. 课程的平均成绩分布图        1. 学生选课数目分布图      1. 学生的平均成绩分布图 | | | |
| 结论分析与体会：    通过不断摸索，以及查找实践，完成了MongoDB数据库的设计，以及数据的导入，基础的增删改查功能，并且实现了一个较为简单的带有操作界面的选课程序，最后对也对给定的数据进行了数据分析可视化的操作。  通过实验也体会到了nosql数据库和关系型数据库的不同，也发现了不同类型的Nosql数据库之间的不同，作为文档型数据库和列数据库从数据库设计上都有很大的不同。因为没有表连接的操作，所以应该尽可能的将关系表进行合并，虽然会产生数据的冗余但是会大大加快数据查询的速度。 | | | |
|  | | | |