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：实验七使用Java做数据统计分析(4学时) | | | |
| 实验学时：4 | | 实验日期： 2018/11/30 | |
| 实验目的   1. 使用Java做数据统计分析(4学时)  实验内容 用API方式，做学生数据统计分析。 重要提醒 将主要程序代码以及运行界面截图保存，用于完成实验报告。 实验要求（在以下要求中选择至少2个）  1. 列出有学生选课的所有课程名称（distinct） 2. 找出平均成绩排名前10的学生 3. 找出选课数目排名前10的学生 4. 找出每位同学的最高成绩以及最高成绩对应的课程名   （5）求每位同学的成绩分布：优秀、良好、合格、不合格的课程门数   1. 求每门课程的选修人数和平均成绩   （7）求每门课程最高成绩以及最高成绩对应的学生姓名  （8）求平均成绩排名前10的课程  （9）求选课人数排名前10的课程 | | | |
| 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 | | | |
| 实验步骤与内容：     1. **实验思路：**   本次实验中给出的数据分析查询，都太过于复杂很难使用简单的查询语句进行查询。所以根据题目的要求，以及后续做数据可视化的需求，可以采用下列两种方案：   1. **JAVA遍历实现**   在之前代码的基础上进行改进，并且可以通过遍历数据求出最大值，平均值等数据来完成实验要求，优点是是灵活，可以转换成任意数据结构，方便后续数据可视化   1. **通过聚合查询：**   MongoDB聚合操作十分强大,可以求解平均值，最大值，排序，并且执行速度快，但是对于太过复杂的查询情况就可能无法满足    根据分析，在本次实验中优先使用聚合查询，在一些较难的题目时使用JAVA实现。其实在MongoDB中也支持MapReduce但是也只能在一个集合中，并且Map映射的结果如果只有一条，则不会执行Reduce,所以很难完成题目的要求，所以本次实验就不采用Mapreduce的方式   1. **MongoDB聚合**   MongoDB中聚合(aggregate)主要用于处理数据(诸如统计平均值,求和等)，并返回计算后的数据结果。有点类似sql语句中的聚合。   1. **聚合管道**   MongoDB的聚合可以将数据按照给定顺序处理，并且管道的操作可以重复。常用的操作符如下：     1. **聚合表达式：**   通过$group操作符分组之后，可以经过统计输出均值，最大最小值等，常用的表达式如下:       1. **java聚合接口：**   Java-mongo-jdbc提供聚合操作的接口：  MongoCollection collection = database.getCollection(colName);  Collection.aggregate(list)     1. **具体题目求解** 2. **列出有学生选课的所有课程名称（distinct）**   本题较为特殊，因为根据数据库的设计，每一堂课都有一个选课学生数组，所以可以通过$where运算符直接查询出：  db.course.find({$where:"this.STUDENTS.length>0"})   1. **找出平均成绩排名前10的学生**        1. **找出选课数目排名前10的学生**      1. **找出每位同学的最高成绩以及最高成绩对应的课程名**   因为需要返回最高成绩对应的课程名，需要访问两个集合的数据，但是聚合只能在一个集合中，所以需要通过Java遍历实现。   1. **求每位同学的成绩分布：优秀、良好、合格、不合格的课程门数**   逻辑过于复杂需要通过java进行遍历处理, 因为数据中存在大于100的分数，所以分数划分为 不合格<60, 合格<80, 良好 <100 ,优秀>100。   1. **求每门课程的选修人数和平均成绩**      1. **求每门课程最高成绩以及最高成绩对应的学生姓名**   同样需要java遍历实现，和问题四相同，因为涉及到多集合操作。     1. **求平均成绩排名前10的课程**        1. **求选课人数排名前10的课程**      1. **具体JAVA代码实现：**      1. **第一题**      1. **第二题**      1. **第三题**      1. **第四题**      1. **第五题**      1. **第六题**      1. **第七题**      1. **第八题**     **实现简单的界面，并且将查询返回的json结果集显示在前端表格界面上。**   1. **运行结果：** | | | |
| 结论分析与体会： | | | |