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。但是在MongoDB中的聚合可以多次使用操作符进行操作，就像让数据通过一个处理管道一样。   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({$where:"this.STUDENTS.length>0"})   1. **找出平均成绩排名前10的学生**   通过聚合可以实现。首先通过$unwind运算符将嵌合选课数组COURSES打开，否则无法操作数组内元素。然后通过$group操作符对SID和NAME进行分组，用$avg求解COURSE.SOCRE的平均值。接着通过$sort,$limit运算符选出Top10，最后用$project运算符将输出重命名。其他通过聚合实现的语句操作基本相同：         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**        1. **第九题**      1. **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。** 2. **运行结果：**   根据题号排序：             1. **通过数据统计将数据可视化：**   数据可视化是数据统计重要的部分，对于题目中所给的数据选课数据通过观察发现数据分布其实非常随意，应该不是真实的数据。所以就通过简单的数据可视化（柱状图，扇形图）展示数据的分布，具体实现一下四个分布：   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     数据的全部来源于上述查询返回的JSON结果，通过简单的数据结构改变，然后通过JFreeChart库将图像画出来，还是可以看出数据的一些变化情况的，很多数据可以看出明显是伪造的，或者是错误的。因为有些数据与其他数据相去甚远。 | | | |
| 结论分析与体会：    通过实验完成全部给出题目并且通过简单数据可视化的方式查看数据的一些特征，并且测试通过，对于本次实验数据分析题目，就是更加复杂的查询，大部分可以通过MongoDB的聚合操作来实现，可以比JAVA处理更加快速有效率，对于特殊过于复杂的一些查询只能通过JAVA代码实现。  MongoDB的聚合操作还是十分强大，对于一些数据分析数据处理的操作还是可以非常直观，快速的得出结果。其中结果集可以进行一些数据可视化的操作，但是因为对数据的真实性的问题无法得出看出数据的一些特点，也很难对数据进行分析。 | | | |