

基于 B/S 的在线考试系统答案更新算法^①

An Algorithm of Answer Update of Online Exam System Based on B/S

彭玉忠 王金才 郝荣霞 (潍坊医学院计算机中心 山东潍坊 261042)

摘要: 基于网络的在线考试系统,是现代考试系统的发展趋势。利用 ASP.NET 和 SQL Server2000 实现的 B/S 结构的在线考试系统,是目前比较先进的在线系统实现方法。通过对数据库中表的独特设计,引入试卷号、答卷号、试题号等字段,利用有限的几个表实现了多种题型的在线考试。利用初始答题表和本次提交的答案的比较,给出了一种答案更新算法,对答题表进行插入和替换,保证了答题表记录的不重复。实验结果表明,该算法可用于多种题型,是一种高效的算法。

关键词: 在线考试系统 答案更新 ASP.NET SQL Server2000 B/S

考试是教学过程中的一个重要环节,通过考试,教师可以了解学生的学习效果,为改进教学提供依据;学生则通过考试了解自己对内容的掌握情况,增强学习兴趣。传统的考试方式,局限性大、资源浪费严重,在网络遍布的今天,将考试系统应用在互联网上是现代考试系统的发展趋势。国外的许多考试,像 GRE、微软认证考试、Cisco 认证考试等已经率先实现了基于网络的考试。所以研制适合具体考试要求并适用于多学科的在线考试系统是非常必要的。

系统用 ASP.NET 平台开发,Web 数据库采用 SQL Server。ASP.NET 提供了迄今为止最先进的 Web 开发平台,它是一个已编译的、基于 .NET 的开发环境,可以用任何与 .NET 兼容的语言 (Visual Basic .NET、C# 和 Jscript .NET) 开发应用程序,可以无缝地与 HTML 编辑器和其他编程工具一起工作^[4]。SQL Server2000 是 Microsoft 公司推出的一种数据库管理和开发工具,是大规模的在线事务处理、数据仓库和电子商务应用的最好的平台^[5]。

1 在线考试系统模型

目前的客户端应用程序模型大致分为两种: C/S 和 B/S。C/S 模型即客户机/服务器结构,在 20 世纪 80 年代出现并得到广泛应用,是一种两层结构的系统,第 1 层是在客户系统上结合了表示与业务逻辑,第 2 层是通过网络结合了数据库服务器,如图 1 所示; B/S 模型即浏览器/服务器结构,是 C/S 模型发展到一定阶段的产物,采用三层结构,它以 Web 为中心,采用 TCP/IP 技术、HTTP 传输协议,客户端通过浏览器访问 Web 以及通过 Web 访问相连的后台数据库,如图 2 所示。B/S 模型的最大优点是:易于安装与维护,大多数计算机已经默认安装有浏览器软件,只需要在服务器端进行部署和维护工作^[1],并且, B/S 模型和客户计算机的操作平台无关。本文涉及的 Web 在线考试系统,是基于 B/S 模型的考试系统^[2,3]。

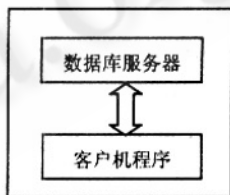


图 1 两层 C/S 结构

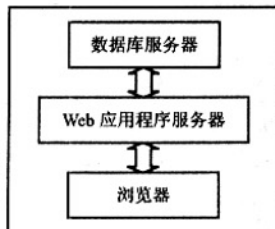


图 2 三层 B/S 结构

① 立项项目:山东省高等学校教学改革立项项目(C05065);山东省高等学校实验技术研究项目(鲁教备字[2005]2号第327项)

2 数据存储结构

本文给出的在线考试系统,利用学生表 Student 保存学生的基本信息和考试信息,如姓名、考号、密码、班级、权限、考场号、场次号、座位号、试卷号、答卷号、考试时间、考试类型、违规类型、交卷类型、考试科目、各种题型得分等。利用题库 QuestionLibrary 保存各种类型的试题,题库包含每道试题的题号、题型、试题内容、标准答案、关联文件、所属章节、难度、本题分值、使用频率等字段。针对每次考试,建立试卷表 PaperTable、试题表 QuestionTable 和答题表 AnswerTable。试卷表包含试卷号、试卷名、出题教师、出题日期、考试日期、考试时间等试卷信息。试题表包含本次考试多套试卷的全部考题,用试卷号表示第几套试卷,用试题号表示题库中的某一道试题。试题表结构如表 1 所示。系统利用题库按本次考试设置的题型、题量、难度、分值、所在章节等信息建立试题表。

表 1 试题表 QuestionTable 结构

含义	字段名	类型	长度	备注
序号	Id	BigInt	8	主键,唯一
试卷号	PaperNo	int	4	表示第几套试题
试题号	QuestionNo	BigInt	8	与题库对应

答题表用于保存本次考试全部考生的所有试题的答题结果,每个考生的每道题插入一条记录。其中,答卷号固定对应某个考生,所有考生的答卷号互不相同。试题号对应试题表中的试题号,学生答案采用 text 类型,用于保存各种题型的答案,由于 text 类型按照答案实际长度存储,节省了存储空间。答题表结构如表 2 所示。

表 2 答题表 AnswerTable 结构

含义	字段名	类型	长度	备注
序号	Id	BigInt	8	主键,自动增长
答卷号	AnswerPaper	BigInt	8	唯一,与学生对应
试题号	QuestionNo	BigInt	8	与题库、试题表对应
学生答案	StudentAnswer	text	-	
本题得分	GetPoint	float	8	

3 更新算法

3.1 更新算法概述

以单选题为例,假设本次考试有试卷 1 和试卷 2

共两套试卷,每套试卷中包含 10 道单选题和 5 道多选题,单选题试题号 SingleChoiceQuestionNo 为 1、2、3、4、5、11、12、13、14、15;多选题试题号 MultiChoiceQuestionNo 为 6、7、8、9、10。试题号存储在试题表中,如表 3 所示,与题库中的试题号相对应。

表 3 试题表 QuestionTable

Id	PaperNo	QuestionNo
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	1	6
7	1	7
8	1	8
9	1	9
10	1	10
11	1	11
12	1	12
13	1	13
14	1	14
15	1	15
16	2	1
17	2	3
18	2	4
19	2	8
20	2	13
...

设学生 A 做答卷 1,其对应答卷号为 1,学生 B 做答卷 2,其对应答卷号为 2。当学生 A 提交 6 道单选题后,学生 B 又提交了 3 道多选题,此时答题表如表 4 所示。其中的 Id 唯一且自动增长,与学生答案提交到服务器的顺序有关。

表 4 答题表 AnswerTable

Id	AnswerPaper	QuestionNo	StudentAnswer
1	1	1	A
2	1	2	C
3	1	5	D
4	1	11	C
5	1	12	D
6	1	15	A
7	2	6	AB
8	2	7	ACDE
9	2	10	BCD

把学生 A 的单选题答案存入数组 AnswerOfFormer, 用于更新算法, 如表 5 所示。

表 5 学生 A 单选题答案表 AnswerOfFormer

题号	1	2	3	4	5	11	12	13	14	15
答案	A	C			D	C	D			A

假设学生 A 再次提交的单选题答案如表 6 所示, 存于数组 AnswerOfSubmit。

表 6 学生 A 提交的单选题答案表 AnswerOfSubmit

题号	1	2	3	4	5	11	12	13	14	15
答案	A	C	D		C	C	D	B	A	B

利用更新算法, 更新后的答题表如表 7 所示。

表 7 更新后的答题表 AnswerTable

Id	AnswerPaper	QuestionNo	StudentAnswer
1	1	1	A
2	1	2	C
3	1	5	C
4	1	11	C
5	1	12	D
6	1	15	A
7	2	6	AB
8	2	7	ACDE
9	2	10	BCD
10	1	3	D
11	1	13	B
12	1	14	A

这里, 更新算法对考生的每次提交的单选题答案与之前提交的答案进行比较, 如果原答题表中已经存在某道题的答案, 则进行替换, 否则插入新记录。保证了答案的唯一性, 为阅卷提供了方便。

3.2 更新算法核心代码

```
string ConnectionString = ConfigurationSettings.AppSettings["SQLConnectionString"];
SqlConnection conn = new SqlConnection(ConnectionString);
for(int i = 0; i < SingleChoiceQuantity; i++)
{
    if(AnswerOfFormer[i] != AnswerOfSubmit[i])
    {
        if(AnswerOfFormer[i].Equals(null))
```

```

    {
        cmdText = "INSERT INTO AnswerTable
            (AnswerPaper, QuestionNo, StudentAnswer)
        Values(" + answerpaper + ", " +
            SingleChoiceQuestionNo[i] + ", " + AnswerOfSubmit[i] + ")";
        SqlCommand MyCommand1 = new
            SqlCommand(cmdText, conn);
        conn.Open();
        MyCommand1.ExecuteNonQuery();
        conn.Close();
    }
    else if(! AnswerOfSubmit[i].Equals(null))
    {
        cmdText = "UPDATE AnswerTable SET Student-
        Answer = " + answerofSubmit[i] + " where
        QuestionNo = " + SingleChoiceQuestionNo[i]
        + " and AnswerPaper = " + answerpaper;
        SqlCommand myCommand2 = new
            SqlCommand(cmdText, conn);
        conn.Open();
        myCommand2.ExecuteNonQuery();
        conn.Close();
    }
}
}
```

代码用 C# 编写, 其中的变量 ConnectionString 为数据库连接字符串, SingleChoiceQuantity 为单选题题量 10, answerpaper 为考生的答卷号, 可从学生表 Student 中读取。代码对学生每次提交的全部单选题答案, 逐一判断前后两次的做答是否一致, 如不一致, 做如下操作: 当初始答题表中某题号对应的答案为空时, 表明前一次提交时没有对该题目做答, 本次提交做答了该题, 使用插入操作将学生答案插入到答题表中; 否则, 表明本次提交的答案对该题目做了更改, 使用替换操作更新答题表。

4 结论

本文给出的基于 B/S 的在线考试系统的答案更新

(下转第 102 页)

(上接第 119 页)

算法,利用数据库中的表保存试卷号、答卷号、试题号等字段,使表间关系清晰,无冗余数据。数据库结构设计新颖,表结构简单,数量少,管理简便,为阅卷和成绩分析奠定了良好的基础。

答案更新算法,不仅对单选题有效,还可以推广应用到多选题、判断题、填空题、文字录入、名词解释、问答题等多种题型,减轻了编程人员的工作负荷。

利用本文的数据存储结构,阅卷过程也不复杂,在与学生表 Student 关联后,可使用查询语句很容易地查询出某个学生的答案,与题库的标准答案比较。比如:查询学生 A 的答题情况,可用 `SELECT QuestionNo, StudentAnswer FROM AnswerTable where AnswerPaper = 1`。

基于 B/S 的在线考试系统答案保存的实现多种多样,本文提出的算法,实践证明是一种高效的算法,已在在线考试系统中得到很好的应用。

参考文献

- 1 张治中. 网络数据库中 C/S 和 B/S 两种模式的特点及其应用. 渭南师范学院学报, 2004, 19(5): 55 - 57.
- 2 张丽萍, 李延衍, 朱斌, 矫桂秋. 基于浏览器方式的考试系统的实现. 大连海事大学学报, 2000, 26(3): 102 - 105.
- 3 陈志国, 张瑾. 网上考试系统的设计与实现. 河南大学学报, 2003, 33(4): 69 - 71.
- 4 章利, 范滨多, 曹乐松, 乔佩利. 基于 ASP . net 动态网站的设计与实现. 信息技术, 2007, 31(1): 68 - 71.
- 5 李代平, 章文. SQL Server 2000 数据库应用基础教程. 北京: 冶金工业出版社, 2001. 18 - 47.