

面向 Online Judge 的查询方法及其实现^①

黄龙军, 范立新

(绍兴文理学院 计算机系, 绍兴 312000)

摘 要: 讨论面向 Online Judge(简称 OJ)的查询方法与实现技术. 重点讨论在本校 OJ 中查询解题进步情况、作业完成情况等存储过程, 及通过 WebClient 对象查询在它校 OJ 的解题情况的方法及其实现.

关键词: Online Judge; 存储过程; 查询; 解题; WebClient

Query Method and Implementation for Online Judge

HUANG Long-Jun, FAN Li-Xin

(Department of Computer Science, Shaoxing University, Shaoxing 312000, China)

Abstract: This paper discusses the query method and implementation for online judge(OJ).It focuses on the stored procedures to query the progress of problem solving situation and the completion of homework in our OJ ,and the query method and implementation by the WebClient object to query the problem solving situation in other OJ.

Key words: online judge;stored procedure;query; problem solving;WebClient

1 引言

OJ 一般指在 ACM/ICPC(国际大学生程序设计竞赛)等各种形式的编程比赛中用来评测参赛选手的程序的正确性与时空效率的程序以及评测程序所依托的网络环境^[1,2]. 用户可以在线提交多种程序设计语言(如 C、C++等)的源代码, OJ 对源代码进行编译和执行, 并通过预先设计的测试数据来检验源代码的正确性^[1,2].

高级语言(C/C++)是高校计算机科学与技术专业及相关专业的专业基础课, 实践性极强. 在教学过程中, 我们以 OJ 为平台进行教学, 提高学生分析、解决问题的能力, 突出学生的主体作用. 为了激发学生的学习兴趣 and 热情、激励学生积极向上并保证教学质量, 我们每周统计学生在本校 OJ 上的解题情况并推出进步排行榜, 统计学生在本校 OJ 上的作业完成情况, 统计学生在其它学校(简称它校)OJ 上的解题情况. 这些工作若由教师手工完成, 将耗费大量的时间和精力, 为方便教师查询、统计, 我们开发了面向 OJ 的查询统计系统, 很好地简化了教师的工作. 查询统计系统采用 B/S(Browser/Server)架构, 具有用户管理、学生信息

维护、作业统计(主要是本校 OJ 上的作业完成情况)、解题统计(包括本校、它校 OJ 的解题情况)、排行榜生成等功能模块.

2 面向本校OJ的查询方法与实现

2.1 查询方法与知识

本校 OJ 的数据库是我们自己设计的, 查询数据可以直接使用数据库中的表等对象, 采用嵌套查询和连接查询的方法.

存储过程是可以一次执行的 SQL 代码块, 作为一个单元存储在服务器端, 并在服务器端执行^[3-5]. 存储过程具有模块化程序设计、提高执行效率、减少网络流量、提供安全机制等优点^[3-5].

PIVOT 和 UNPIVOT 是 SQL Server 2008 系统的关系运算符, 提供了一种把列数据转换为行数据的方式. PIVOT 运算符把表达式某一列中的唯一数据转换为输出中的多个列^[5]; UNPIVOT 运算符则相反, 将列转换为行. 我们规定, 高级语言的教学内容分为 7 章, 章名分别为: 绪论、基础知识、控制结构、数组、函数、指针、

① 基金项目:绍兴文理学院第三批课程教学模式改革项目

收稿时间:2013-01-02;收到修改稿时间:2013-01-28

结构; 通过 PIVOT 运算, 可以把每个学生每章完成情况的 7 条记录转换为 1 条记录(把各章名作为列名).

2.2 数据对象

在查询解题情况并生成进步排行榜及统计作业完成情况的存储过程中, 主要用到的对象有用户表([User])、问题表(Question)、答案表(Answer)、分类表([Group])、学生表(Student)、作业视图(Task), 其结构与字段说明如下:

User(id,username,sno,submit,accepted,solved)

各字段的含义为: 序号, 用户名, 学号, 提交数, 正确数, 解题数.

Question(id,pid,title,content,submitCount,acceptedCount)

各字段的含义为: 序号, 问题编号, 标题, 内容, 提交数, 正确数.

Answer(id,questionid,username,status,addtime,code)

各字段的含义为: 序号, 问题编号, 用户名, 答案状态, 提交时间, 代码.

Group(id,name,classify)

各字段的含义为: 序号, 名称, 类别号.

Student(sno,name,id,hdojId,zojId)

各字段的含义为: 学号, 姓名, 本校帐号, 杭电帐号, 浙大帐号.

Task(id,questionId,chapter,chapterName)

各字段的含义为: 序号, 问题编号, 章号, 章名.

2.3 具体实现

2.3.1 存储过程 getProgress

存储过程 getProgress 的功能是获取本校账号范围从@sun到@eun, 时间范围从@st到@et的学生进步排行榜, 具体定义如下:

```
create procedure getProgress(@sun nvarchar(20),
    @eun nvarchar(20),@st datetime, @et datetime) as
begin
    set nocount on
    create table #tn --建立临时结果表
    ([name] nvarchar(10), username nvarchar(20),
        progress int default 0, total int default 0)
    insert into #tn
        --把以下嵌套查询结果插入临时结果表
    select ( (select [name] from student where
        id=T.username) [name], --获得姓名
```

```
T.username, --获得账号(用户名)
--查询时间范围之前未正确过的解题数量
(select count(distinct QuestionId)
    from Answer,Question,[Group]
        --连接查询
    where Answer.QuestionId=
        Question.Id and
        Question.GroupId=[Group].Id
        --连接条件
    and [Group].classify=1
        --作业题目 classify 为 1
        --练习题目 classify 为 0
    and status=1
        --代码正确则 status 为 1,
        --否则为 0
    and Answer.addTime
        between @st and @et
        --时间范围
    and Answer.username=
        T.username
        --(相关)嵌套查询条件
    and Answer.QuestionId
        not in(
        --选择以前未正确过的题号
        select QuestionId
        from Answer
        where status=1 and
        Answer.username=T.username
        and addTime < @st
        )
    ),
    solved --解题总数
from [user] as T
where T.username between @sun and @eun
        --账号范围
group by T.username, solved --分组
-- rank()是排名函数, 详见文献[3]
select rank() over (order by progress desc)
    [Rank], * from #tn
drop table #tn --删除临时表
end
```

在 SQL Server 中执行该存储过程的语句如下:

```
exec getProgress 'usx12201', 'usx12205',
                '2012-12-15','2012-12-22'
```

通过该语句的执行, 获得本校 OJ 上帐号范围从 usx12201 到 usx12205 的学生在一周时间范围内(从 2012-12-15 到 2012-12-22)的进步排行榜, 如表 1(表中略去学生姓名列)所示:

表 1 进步排行榜

Rank	username	progress	total
1	usx12201	14	108
2	usx12205	11	205
2	usx12202	11	241
4	usx12204	8	121
5	usx12203	7	112

2.3.2 存储过程 getDone

存储过程 getDone 的功能是统计本校 OJ 账号范围从 @sun 到 @eun 的学生的各章作业完成情况, 具体定义如下:

```
create procedure getDone(@sun nvarchar(20),
                        @eun nvarchar(20)) as
begin
    set nocount on
    select [Sno] 学号,
           [Name] 姓名,
           ChapterName ,
           floor(cast((Solved*100.0/Total) as
                      decimal(5,1))) [Percent]
           --每章作业完成百分率
    into #tn --用查询结果生成临时表
    from
    (select (select [Sno] from student where
                 id=A.username) [Sno], --获得学号
            (select [name] from student where
                 id=A.username) [Name], --获得姓名
            A.Username, Chapter, ChapterName,
            --获得账号, 章号, 章名
            (select count(*) from Task where
                 chapter=T.chapter) Total,
            --获得每章作业总数
            count(distinct A.Questionid) Solved
            --获得每章完成题数
```

```
from Answer as A, Question as Q, task as T
where A.QuestionId=Q.Id and
```

```
Q.Qid=T.questionId --连接查询
```

```
and status=1 --正确代码标记为 1
```

```
and A.username between @sun
```

```
and @eun --账号范围
```

```
group by A.username, chapter, chapterName
```

```
) as Result
```

```
order by chapter,[Percent] desc,[Sno] --排序
```

```
--通过 pivot 运算把每个学生各章的行数据
```

```
--转换为列数据
```

```
select *
```

```
from #tn pivot
```

```
(sum([Percent]) for chapterName in(
```

```
绪论, 基础知识,控制结构,数组,
```

```
函数,指针,结构)) tab --tab 为别名
```

```
drop table #tn --删除临时结果表
```

```
end
```

在 SQL Server 中执行该存储过程的语句如下:

```
exec getDone 'usx12201','usx12205'
```

通过该语句的执行, 获得本校 OJ 上帐号范围从 usx12201 到 usx12205 的作业完成情况, 如表 2(表中略去学生姓名列)所示:

表 2 作业完成情况

学号	绪论	基础知识	控制结构	数组	函数	指针	结构
12136201	100	100	100	83	87	100	35
12136202	100	100	100	100	100	100	100
12136203	100	100	94	71	87	100	57
12136204	100	100	94	77	100	100	71
12136205	100	100	100	100	100	100	100

在应用系统的相关网页中显示进步排行榜及作业完成情况, 可以采用文献[3]中涉及的“通过 ADO.NET 调用存储过程”的方法, 不再赘述。

3 面向它校OJ的查询方法与实现

3.1 查询方法

因为我们不可能直接到它校 OJ 的数据库中获取数据, 所以面向它校 OJ 的查询方法是通过它校 OJ 与排名相关的网页信息来获取解题数据。ASP.NET 的

WebClient 类提供了将数据发送到由通用资源标识符 (Uniform Resource Identifier, 简称 URI) 标识的资源及从这样的资源接收数据的常用方法. 使用 WebClient 类对象下载网页并转换为字符串信息, 通过分析网页信息并进行字符串处理, 便可以很容易获取学生的解题数量.

3.2 具体实现

我们编写的 C# 函数 `getSolvedByUid` 的功能是在参数 `urlStr` 标识的网页中, 按学生账号参数 `uid` 及查找目标字符串 `target` 参数和分隔字符 `splitCh` 参数获取解题数量, 具体定义如下:

```
protected string getSolvedByUid(string urlStr, string
uid, string target, int length, char splitCh) {
    string url = urlStr + uid;
    //创建 WebClient 对象
    WebClient wb = new WebClient();
    //下载网页信息
    byte[] pagedata = wb.DownloadData(@url);
    //信息转换为字符串
    string t = Encoding.Default.GetString
        (pagedata);
    //字符串处理, 获取解题数量
    int index = t.LastIndexOf(target);
    if (index == -1) return "0";
    t = t.Substring(index);
    string res = "";
    for (int i = length; t[i] != splitCh; i++)
        res += t[i];
    return res;
}
```

调用该函数的语句如下:

```
string hduSolved = getSolvedByUid(
```

```
"http://acm.hdu.edu.cn/userstatus.php?user=",
hdojId, "Problems Solved", 37, '<');
```

```
string zjuSolved = getSolvedByUid(
    "http://acm.zju.edu.cn/onlinejudge/" +
    "showUserStatus.do?handle=", zojId,
    "AC Ratio:", 44, '/');
```

通过上面语句的执行, 获得一个学生(其杭电账号为变量 `hdojId` 的值, 浙大账号为变量 `zojId` 的值)在杭州电子科技大学 OJ 和浙江大学 OJ 上的解题数量, 各自存放在一个字符串变量中, 以便于后续操作. 通过遍历 `Student` 表, 把每个学生记录的杭电账号、浙大账号作为参数, 便可以批量查询所有学生的解题数量.

4 结语

在查询统计学生在 OJ 上的解题情况、作业情况时, 通过采用本文讨论的方法, 避免了机械的人工操作, 简化了教师的工作, 所用的方法有较好的实用价值; 本文讨论的存储过程的实现方法及相关知识对于数据库学习者和研发人员有一定参考价值.

参考文献

- 1 庄奇东, 王键闻, 张楠, 张爽, 任娜. Online Judge 系统的优化. 计算机系统应用, 2011, 20(8): 115-121.
- 2 黄龙军. 游标在 Online Judge 中的应用. 绍兴文理学院学报, 2012, 32(8): 26-29.
- 3 黄龙军. 应用存储过程实现数据分页. 计算机系统应用, 2012, 21(12): 186-189.
- 4 黄龙军. 存储过程技术在网络考试系统 SQL 注入攻击防御上的应用. 计算机系统应用, 2013, 22(1): 103-106.
- 5 闪四清, 邵明珠. SQL Server 2008 数据库应用实用教程. 北京: 清华大学出版社, 2009. 221.

(上接第 57 页)

参考文献

- 1 步山岳, 张有东. 计算机信息安全技术. 北京: 高等教育出版社, 2005.
- 2 樊金一, 傅世林, 邸书灵, 等. PowerBuilder 6.5 实用教程. 北京: 科学出版社, 2000.
- 3 殷红先, 张为民. PowerBuilder 实用教程. 北京: 电子工业出版社, 2009.

社, 2009.

- 4 <http://www.itownet.cn/index.shtml>.
- 5 Yan DM, Dan ZG. ZigBee-based Smart Home system design. Advanced Computer Theory and Engineering (ICACTE), 2010, 3rd International Conference, 2010, 2: 650-65.