正则表达式在数据库查询中的应用

李 旻， 陈和平
（武汉科技大学 信息科学与工程学院，湖北 武汉 430081）

摘 要：SQL 语句在数据库查询中具有非常重要的地位，但是标准的 SQL 语句在复杂的数据库查询中却存在着诸多不足。而正则表达式有着强大的查询功能，通过对正则表达式特殊字符以及数据库查询语言中谓词的分析，提出了将正则表达式运用于数据库查询当中的新查询方法，并对该方法在实际查询应用当中会遇到的几个普遍问题进行了探讨。从而证明该方法不仅可以降低查询语句的复杂程度而且还能简化对出错语句的修改工作。

关键词：正则表达式； SQL； 数据库查询； 模式； 模式匹配； 特殊字符
中图法分类号：TP311.132.3 **文献标识码**：A **文章编号**：1000-7024 (2006) 12-2303-03

Application of regular expression in database query

LI Min, CHEN He-ping
(College of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China)

Abstract: SQL sentences play an important role in database query. But there are many shortages while doing some complicated queries by the traditional SQL sentences. While the regular expression has the strong power in query, the special characters of the regular expression and the predication in the database query are analyzed. Then a new query method basis on regular expression is presented, and some common problems in application are probed into. With the application of this method the complication of the query sentences is reduced and the modifying of the error SQL sentences is simplified.

Key words: regular expression; SQL; database query; pattern; pattern matching; special character

0 引 言

在数据库字段中查找所需文本数据是应用程序编程的常见工作。标准 SQL 语句采用谓词 LIKE 的方式来进行模式匹配，但该方式限制较多，使用起来并不是那么尽如人意。

假设要从职工表 Table1 中查询满足条件“姓名字段以字母 A、B、C 或 D 中任一个字母打头且以字母为 P 结束”的所有记录，其标准的 SQL 查询语句如下

```
SELECT * FROM Table1
WHERE name like 'A%P' or name like 'B%P' or name like 'C%P' or name like 'D%P'
```

该查询要求十分明确且并不太复杂，上述查询语句的执行结果也完全正确，但 SQL 语句却显得有点冗长。语句中的百分号(%)可代表任意多个字符。倘若要对姓名字段的长度加以限制，则势必会增加 SQL 语句的复杂度。如果在数据库查询中仅靠采用标准的 SQL 语句来实现各种查询要求，复杂的查询语句不仅难以编写，而且还有可能降低数据库的查询速度。一旦出错，则更是难以进行排查和修改。

对于上述问题，选择使用正则表达式将不失为一种较好的解决方案。运用正则表达式可以在给定的字符串中查找特定的模式，使用转义字符序列来描述具有特定含义字符及字符的集合。在进行数据库的各种查询操作过程中，灵活运用正则表达式的特殊功能，将会有助于降低查询语句的复杂度并提高编程工作的效率。

1 正则表达式

正则表达式是一种可用于字符串模式匹配和替换的强有力工具。最简单的模式就是一个所要查找的字符串。模式在处理文档工作中是非常普遍的，语句 SELECT * FROM Table1 WHERE name = 'Tom Smith' 中的字符串 'Tom Smith' 就是一个简单的模式。实际应用中用户所要查询的是更为一般、更为通用的模式。

表 1 给出了部分正则表达式的特殊字符及其说明。表 2 给出了在数据库中谓词的语法规则。

2 使用方法

正则表达式提供了谓词 REGEXP_LIKE 来进行模式的匹配。同样完成上述查询工作的带正则表达式的 SQL 语句如下：

```
SELECT * FROM Table1
WHERE REGEXP_LIKE (name,'^[A-D]P$')
```

收稿日期：2005-04-18。
作者简介：李旻 (1979－)，女，湖北武汉人，硕士研究生，研究方向为计算机网络与数据库应用技术； 陈和平 (1956－)，男，湖北武汉人，硕士，教授，研究方向为计算机网络与数据库应用技术。

表 1　部分正则表达式的特殊字符及说明

字符	说 明
.	表示任何字符
^	表示字符串的开始
?	匹配 0 次或 1 次
{m}	正好匹配 m 次
$	使某一表达式定位于一行的结尾
[:alpha:]	仅限字母字符
[:punct:]	仅限标点字符
[:space:]	空白字符(禁止打印),如回车符、换行符、竖直制表符和换页符

表 2　部分正则表达式在数据库中的语法规则

语法	说明
REGEXP_LIKE(source_string, pattern [, match_parameter])	source_string 支持字符数据类型(CHAR、VARCHAR2、CLOB、NCHAR、NVARCHAR2 和 NCLOB,但不包括 LONG)。pattern 参数是正则表达式的另一个名称。match_parameter 可选参数。
REGEXP_INSTR(source_string, pattern [, start_position [, occurrence [, return_option[, match_parameter]]]])	该函数查找 pattern,并返回该模式的第一个位置。可以指定想要开始搜索的 start_position。occurrence 参数默认为 1,除非指定要查找接下来出现的一个模式。return_option 的默认值为 0,它返回该模式的起始位置 值为 1 则返回符合匹配条件的下一个字符的起始位置。
REGEXP_REPLACE(source_string, pattem[, replace_string [, position[,occurrence, [match_parameter]]]])	该函数用一个指定的 replace_string 来替换匹配的模式,从而允许复杂的"搜索并替换"操作。

其中:字符'^'表示定位于一行的开始处,而字符'$'表示定位于一行的结尾处;操作符 REGEXP_LIKE 与 LIKE 很相似,主要区别在于:LIKE 操作符将查询模式与整个字段的值相匹配,而 REGEXP_LIKE 操作符则可在字段中的任意处开始匹配所要的模式。

由上可见,通过使用正则表达式,原 SQL 查询语句被大大简化了。对于一般较长的 SQL 查询语句,在运用了正则表达式之后将变得简单易写,而对于那些更为复杂的 SQL 查询语句,运用正则表达式之后则更会受益匪浅。

除了以上的谓词 REGEXP_LIKE 以外,数据库还提供另外 3 个:REGEXP_INSTR、REGEXP_SUBSTR 和 REGEXP_REPLACE。它们在用法上与一般的 SQL 谓词 INSTR、SUBSTR 和 REPLACE 用法相似,但是它们使用正则表达式代替了老的百分号(%)和通配符(_)字符。

下面,以验证电话号码格式的数据库应用为例,对正则表达式在数据库查询中的运用方法作进一步介绍。我们不妨设 phone 的格式为:111-222-3333。

首先,创建一个表 userinfo,然后,针对已建好的表及测试数据编写数据库存储过程,最后,以 VC 编程为例说明如何实现基于正则表达式的数据库查询。

2.1　创建数据库表并插入数据

```
CREATE TABLE userinfo(username VARCHAR2(20) , password VARCHAR2(40), email VARCHAR2(50) NOT NULL,phone VARCHAR2(12) NOT NULL, hobbies VARCHAR2(2000))

INSERT INTO USERINFO (USERNAME, PASSWORD, EMAIL, PHONE, HOBBIES) VALUES ('Jason', 'steve123', 'Jason@Jason.com', '515-123-4567', 'football, writing, Cricket',);

INSERT INTO USERINFO (USERNAME, PASSWORD, EMAIL, PHONE, HOBBIES) VALUES ('Neena', 'neena88', 'Neena@Neena.com', '515-123-4568', 'foot ball, guitar ',);

INSERT INTO USERINFO (USERNAME, PASSWORD, EMAIL, PHONE, HOBBIES) VALUES ('Lex ', 'Lex001', 'Lex@email.com', '515-123-4569', 'reading,singing',);

INSERT INTO USERINFO (USERNAME, PASSWORD, EMAIL, PHONE, HOBBIES) VALUES ('David', 'dav007', 'David@email.com', '590-423-4569', 'base ball, cricket, music',);
```

2.2　创建存储过程

(1) 用于格式检查的存储过程。

```
CREATE OR REPLACE PROCEDURE CHECK_PHONE (phone IN VARCHAR2) ASisValid BOOLEAN;
BEGIN
  isValid := REGEXP_LIKE(phone,'^([[:digit:]]{3}-[[:digit:]]{3}-[[:digit:]]{4})$');
  IF (NOT isValid ) THEN
   raise_application_error('Phone number is not valid');
  END IF;
 END IF;
```

该存储过程用于判断数据库字段 phone 的正确性。正则表达式 REGEXP_LIKE (phone,'^([[: digit:]] {3}-[[: digit:]] {3}-[[: digit:]]{4})$')具体给出了判断规则,即电话号码的格式由任意 3 个数字开头后接连字符,后再接 3 个任意数字,再由连字符接 4 个任意数字。可以看出,在 REGEXP_LIKE 表达式中由[:digit:]表示任意数字,而由{3}表示重复的次数,最后'$'表示结尾符。这一判断规则若改由标准的 SQL 语句来实现其复杂度可想而知。用同样的方法还可以检验 E-mail 的格式等。

可以在程序中直接写入 SQL 语句来执行查询工作

```
...
try {
   conn = connManager.getConnection();  // 获得连接
   CallableStatement proc = null;
//检验 E-mailde 格式
    regFrame.addMessage("");
    regFrame.addMessage("Executing Regular Expression to validate Email...");
    regFrame.addMessage(" REGEXP_LIKE('"+ user.getEmail()+"'",'([a-z]+|[0-9]+)@[[:alpha:]]+[.]COM','i')");
    proc = conn.prepareCall("{call check_email(?)}");
    proc.setString(1,user.getEmail());
    proc.execute();
}
...
```

(2) 用于查询的存储过程。

查询要求:找出所有记录中 hobbies 与音乐有关的人员名

单。分析前面的插入数据可知 hobbies 字段的数据模式并不规范，例如：music、violin、guitar 和 singing 等都与音乐有关。这给采用 SQL 语句编程造成了不便。但如果运用正则表达式来表示上述查询要求，问题将简单得多。对应该查询的存储过程如下

```
CREATE PROCEDURE find_hobbies AS
DECLARE
username VARCHAR2(30);
hobbies VARCHAR2(500);
CURSOR hobbies_cur IS SELECT username, hobbies FROM userinfo WHERE REGEXP_LIKE(hobbies, 'music|violin|guitar|sing(er|ing)','i');
BEGIN
LOOP
FETCH hobbies_cur INTO username, hobbies;
EXIT WHEN hobbies_cur%NOTFOUND;
DBMS_OUTPUT.PUT_LINE( username || '  '|| hobbies);
END LOOP;
CLOSE hobbies_cur;
END;
```

其中，正则表达式 REGEXP_LIKE (hobbies, 'music|violin|guitar|sing (er|ing)','i') 负责完成以上查询要求。其中'|'表示替换，在列出的几项中可以相互替换。并且该字符串不区分大小写。

2.3 调用存储过程

在实际应用中，可以将正则表达式直接应用于数据库的查询语句中，或者调用存储过程，两者效果完全相同。

在 VC 编程过程中，可按下列步骤调用已创建好的存储过程

```
_CommandPtr pCmd;
pCmd.CreateInstance("ADODB.Command");          //创建 ADODB 命令
pCmd->ActiveConnection=TEST1;                  //连接数据库
pCmd->CommandText=" find_hobbies "             //将存储过程名负值给命令
pCmd->Execute( NULL, NULL, adCmdStoredProc); //执行存储过程
```

运行结果如下

```
username        hobbies
-------------------------------------------------------------
Neena           foot ball, vguitar
Lex             reading，singing
David           base ball, cricket, music
```

其查询结果与预期结果完全相同。

3 结束语

数据库查询，特别是针对大量数据的复杂查询对于编程的难易程度以及程序的运行速度至关重要，正则表达式是一种用来进行字符串模式匹配的强有力工具。与一般的 SQL 查询语句相比，用正则表达式进行数据库查询不仅可以降低查询语句的复杂程度，而且还能简化对出错语句的修改工作。

从本文所给出的例子可以看出，正则表达式可以让用户通过使用一系列的特定字符构建查询模式，其所提供的强大的模式匹配功能大大简化了繁琐的 SQL 查询语句的编写工作。当然，要运用正则表达式写出高质量的 SQL 语句还需要反复运用和精通正则表达式的所有特殊字符，也只有通过实际使用的过程才能真正体会到运用正则表达式在数据库查询中的优越性。

参考文献：

[1] David Medintes,赵卫红. PHP3 程序设计[M].北京:机械工业出版社, 2000.130-136.

[2] Friedl J E F. Mastering Regular Expressions[M]. *USA:O'Reilly and Associates Inc*, 1997.1-4,31-67.

[3] Jonathan Gennick,Peter Linsley.Oracle regular expressions[M]. *USA:O'Reilly and Associates Inc*, 2003.4-7.

[4] Alice Rischert. Writing better SQL using regular expressions [EB/OL]. *http://www.oracle.com/technology/oramag/webcolumns/*2003/*techarticles/rischert_regexp_pt1.html*.

[5] Larry Wall, Jon Orwant, Tom Christiansen. Programming perl [M].*USA:O'Reilly and Associates Inc*, 2000.

[6] Tony Stubblebine. Regular expression pocket reference [M]. *USA:O'Reilly and Associates Inc*, 2003.2-13.

(上接第 2302 页)

业过程中，不断提高企业的开发创新能力、企业经营管理能力和竞争力。而把 Web Services 技术与企业信息化的思想相结合，形成基于 Web Services 的企业信息化的发展模式，将会为企业带来巨大的效益，也会促使企业健康可持续的发展。

参考文献：

[1] 李劲. 动态电子商务的 Web 服务 [M]. 北京: 清华大学出版社, 2002.

[2] 谢铉洋, 谢荣传. 基于 XML 和移动 Agent 的企业应用集成体系结构[J]. 计算机应用, 2002,22(5):24-26.

[3] 孙晋文,肖建国.企业应用集成与基于 Web Services 的构架应用[J].计算机工程与应用, 2003,(21):205-208.

[4] 许科峰,高建民.基于 Web Services 的企业应用集成技术及实现[J].计算机应用, 2004,25(3):155-157.

[5] 黄卓,张涛.基于 Web Services 的分布式模型管理方法研究[J]. 计算机工程与设计, 2004,(3):379-380.

[6] 赵慧娟,王淑营.面向中小企业信息化建设的 ASP 服务平台[J]. 计算机集成制造系统——CIMS, 2004,(11):1441-1445.