

2012 “种子杯” 程序设计大赛初赛说明文档

SQL 语句的解析与实现

队名 xxx	光学与电子信息学院
徐哲钊	电子科学与技术 1004 班
徐洁	电子科学与技术 1003 班

目录

SQL 语句的解析与实现.....1

一、编译环境说明.....3

 1.1 开发测试环境3

 1.2 源代码目录3

二、程序结构.....3

 2.1 设计结构4

 2.2 模块功能7

三、程序的亮点.....9

四、程序的完成情况.....9

五、感受.....10

一、编译环境说明

本部分对程序开发使用的编译器，开发环境作介绍

1.1 开发测试环境

程序在 visual studio 2008 英文专业版和 visual studio 2010 中文版编译器的环境下开发，其编译运行方式比较简单，不做详细介绍。

1.2 源代码目录

为了使文件更加规范、整齐，也不至于太繁琐，把功能相近的函数组合在一起。最后得到的工程目录下面的文件包括

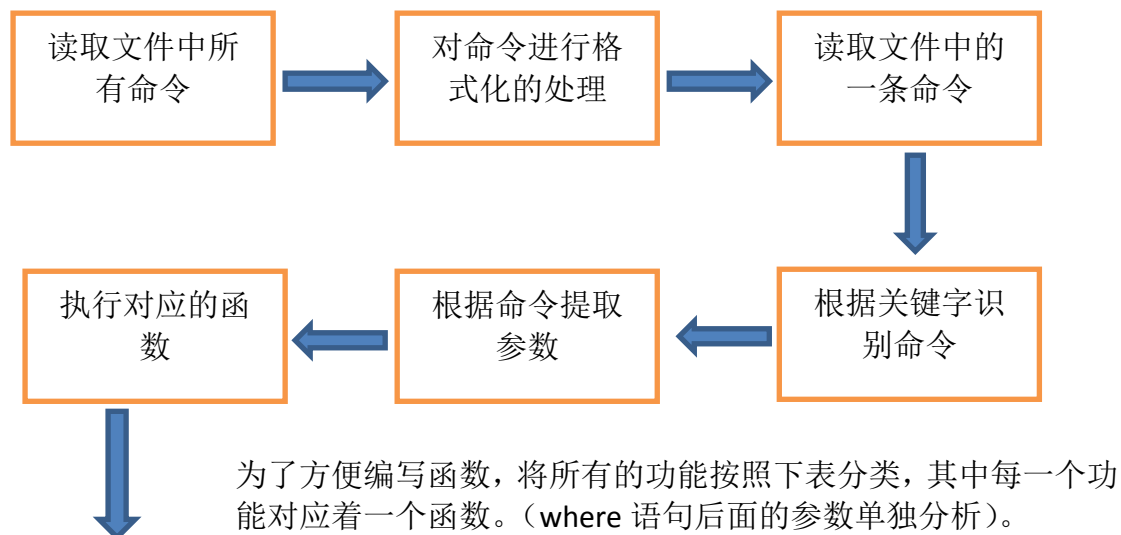
名字	大小/kb	所含函数的主要作用
SeedCup.h	6	声明函数、全局变量、定义数据结构
add.cpp	2	添加记录
auxiliary.cpp	20	排序和辅助函数
create.cpp	3	建立数据库、创建表、添加列
del.cpp	5	删除数据库、表、记录
exe_command.cpp	23	语法解析
gram_parse.cpp	8	命令的识别与匹配
main.cpp	1	从文件读入命令并将结果输出到文件
reanme.cpp	2	重命名数据库
select.cpp	8	select 语法解析
show.cpp	3	显示数据库名、表名、列名
str_parse.cpp	9	格式化字符串
update.cpp	7	修改数据类型
use.cpp	1	指定数据库

二、程序结构

本部分详细介绍了程序的设计构架，各个模块的设计，以及各个模块的功能

2.1 设计结构

2.1.1 总体思路



命令分类		可选参数	对应的函数的功能
create	database	无	建立数据库
	table	无	建立表
alter table	add	无	添加列
	drop column	无	删除列
	alter table	无	修改数据类型
truncate	table	无	删除表保留表的结构
use		无	使用数据库
drop	参数个数为 1	无	删除数据库
	参数个数为 2	无	删除表
rename	table	无	重命名表
	database	无	重命名数据库
update	set where	无	修改表中数据
delete	from where	无	删除表中数据
insert into	values	无	添加记录
	() values	无	指定添加记录的列
show	database	无	打印数据库
	database_name table		打印指定数据库中的表
	table		打印当前数据库中的表
	column	order by	打印指定表的所有列
select	不含 where	order by	打印指定数据

	含 where		打印指定数据
--	---------	--	--------

含有 **where** 语句时，操作符种类比较多，将不同类型的操作符分在不同的函数中实现。

	操作符	对应的函数名
含有 where	==, ~=, <, >, <=, >=	select_where_cmp
	between	select_where_between
	like	select_where_like
	and	select_where_and
	or	select_where_or

2.1.2 程序框架

为了更加方便地描述数据，采用链表的形式来表示数据库和表、记录的连接和存储方式。下面分别介绍记录，表，数据库的成员。

记录

每一个记录对应一个 **record** 类型的结构体，它含有三个成员变量

类型	变量名	作用
int	int_x[20];	存储 int 类型的数据
float	float_x[20];	存储 float 类型的数据
string	string_x[20];	存储 text 类型的数据

表

每一张表对应一个 **table** 类型的结构体，它含有四个成员变量

类型	变量名	作用
list<record>	table_c	链表，表示它所含的记录
string	table_name	表示这个表的名字
string	property[50]	每一个记录所含变量的名字
int	col_type[50]	表示表中所含记录的每一个成员变量的数据类型。(可能为 0: NONE 1: int 2: float 3: string(即 TEXT)).

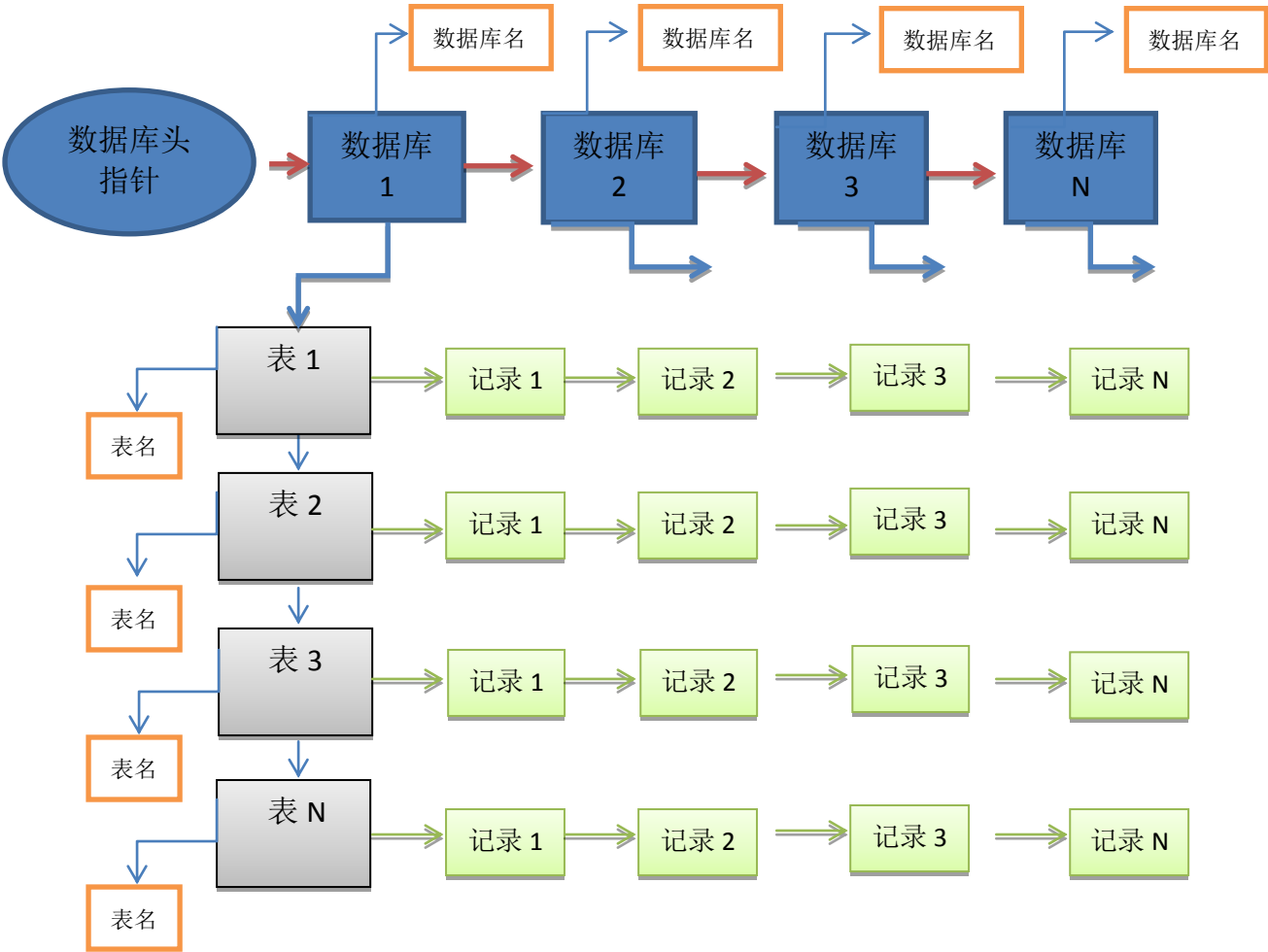
数据库

每一个数据库含有两个变量

类型	变量名	作用
list<table>	db_c	一个链表，表示它所含的所有表
string	db_name;	表示这个数据库的名字

2.1.3 数据存储结构

由上述说明，整个程序的构架即如下图所示



2.1.4 处理字符串

为了方便后续对字符串的提取，我们首先对字符串进行了处理，其中处理的规则如下表所示

命令所含字符	转换规则	转换以后
单个“，”，换行键	逗号转换为空格	空格
<,>==,(,]	在左右添加空格	“ < ” 等等
引号中间出现空格如' '	替换为%	'%'
出现相连的逗号括号，逗号和逗号如(,,)	在中间添加空格	(, ,)

2.1.5 命令识别

命令识别分为三步：

- 1) 预先处理所有命令。利用字符串的转换规则对所有命令进行预处理，以达到便于提取和识别的目的。
- 2) 根据分号提取出每一条命令；各个命令都有特定的关键字，所以直接根据关键字就可以识别命令是哪一条。例如在一条命令字符串中分别查找字符串子串 **Create** 和 **Database**，若存在则说明为应该执行的命令为 **Create Database database_name** 建立数据库。
- 3) 识别出所执行的命令类型以后，根据命令的特征提取出相应的参数。

2.2 模块功能

文件中包含的函数为 70 个，不一一列举，就其中比较典型和重要的功能实现的方案给出说明和介绍。

2.2.1 数据库定义部分

关键功能

ALTER TABLE

介绍如何在已有的表中添加、修改删除列。

record 结构体中存储记录的数值，在 **table** 结构体中的 **string** 类型的变量 **properity[50]** 存储的则是变量的名字（字符串），**table** 结构体中的 **col_type[50]** 中的数值代表的则是每个变量的数据类型；其中 0: NONE 1: int 2: float 3: string(即 TEXT)。

操作	实现方法
添加列	只需将列名保存在 <code>property[50]</code> 中
删除列时	将该列对应的 <code>property[50]</code> 中的数据删除， <code>col_type[50]</code> 中数值删除
修改数据类型时	先修改 <code>col_type[50]</code> 中对应的数值，然后进行数据转换

2.2.2 数据操作部分

这里选择比较典型的功能详细介绍。

SELECT

"*"号的处理：select 语句的第一个参数可能是*,也可能是需要选择的列名，为了统一，将*转换成为该表的所有列的列名，将列名存储在 `string` 类型的数组中。

表名：from 后面的参数制定了选择的表

where 语句的处理：如果含有 where 则分情况实现

操作符为 `==, >=, <=, ~=, >, <` 为一种情况；操作符为 `between, like, and, or` 时各分一种情况分别实现。根据相应的函数可以实现。

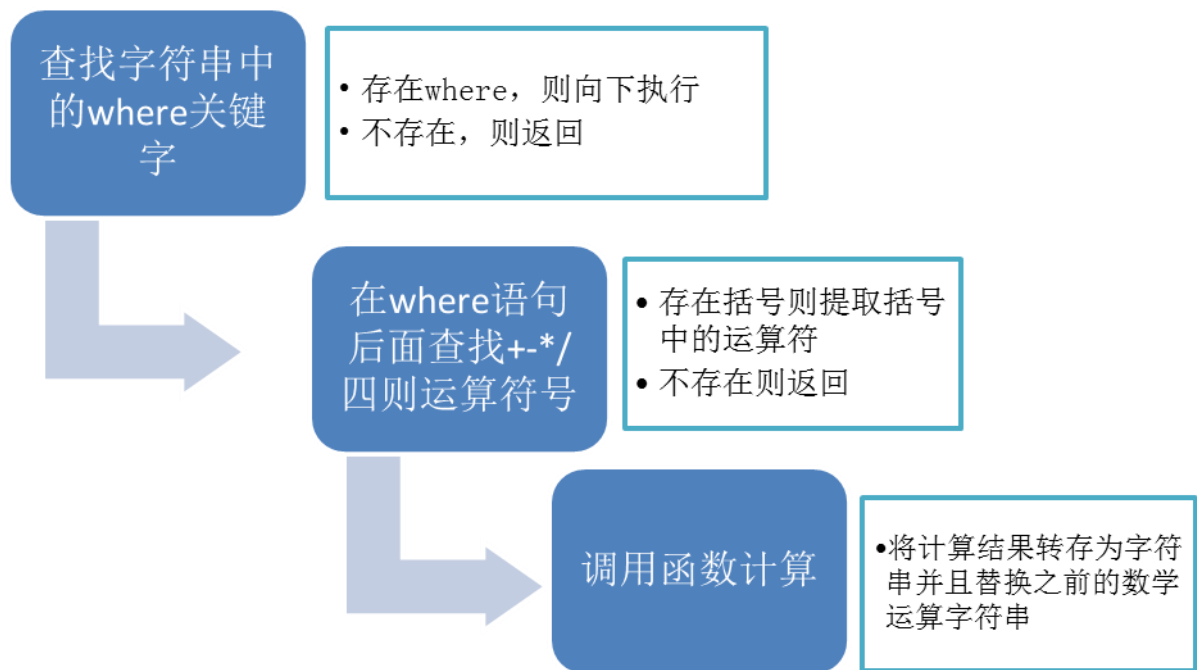
order by 的处理：使用 `string` 数组保存 order by 语句，如果不含有则对应的 `string` 为空按照默认排序，含有 order by 则提取参数 `desc/incr`，分别按照降序和升序排列。

打印数据：先遍历表中所有的记录，找到符合要求（where 语句）的记录，然后根据选中的列来显示。

2.2.3 复杂的数学运算

为了识别一条命令是否含有复杂的数学运算，通过判断 `where` 关键字后面是否含有加减乘除的符号，如果含有则提取 where 语句中的运算符保存到 `string` 的字符串中，然后调用函数处理，得到结果以后，将数值转换为字符串并且替换原来的复杂数学运算字符。

具体实现方案流程为



三、程序的亮点

- 1) 使用链表存储, 接近实际记录的存储方式, 便于理解
- 2) 对命令进行预先处理, 按照方便提取和识别的原则, 统一每一条命令的格式
- 3) 将命令的识别、命令的参数提取、命令的执行分开, 使程序结构更加清晰明了
- 4) 实现了复杂的数学运算, 并且能够处理除数为 0 的情况
- 5) 程序的容错性非常好, 能够识别大部分的语法和输入错误
- 6) 实现了通配符*, ? 的匹配
- 7) 实现了 and 和 or 的逻辑功能

四、程序的完成情况

实现了试题文档中的基本功能, 如果命令正确则能够成功执行得到结果; 但是对于附加分数, 只实现了复杂的数学运算这一项。另外虽然程序能够识别大部分的错误, 但是考虑到实际的错误种类情况太多, 程序可能会在一些错误输入的

情况下意外终止，这也是程序需要改进但是却很难完全改正的地方。

五、感受

对于一般只学过 c 语言的同学，也许他们完全不懂 c++，对数据结构一窍不通，不知道如何使用 stl... 但是我们两同学却很是与众不同：老师没讲数据结构，我们自学；c++不懂，我们上网反复查阅资料；stl 如何使用，我们查阅大量的例子修改复杂的代码。种子杯软件设计大赛，没有软件设计基础的我们却包含兴趣与激情，敢于试一试。初略计算一下，程序代码超过 3000 行，函数接近 70 个，尽管程序中很多代码很冗杂繁琐，函数功能不能称得上完善，但是我们善始善终坚持了下来，最终实现了绝大部分的功能。