

2012 “种子杯”

初赛试题

2012/10/23

1. 题目概述

结构化查询语言(Structured Query Language)简称 SQL ,结构化查询语言是一种数据库查询和程序设计语言,用于存取数据以及查询、更新和管理关系数据库系统。

结构化查询语言是高级数据操作语言,不要求用户指定对数据的存放方法,也不需要用户了解具体的数据存放方式,所以是一种高效的数据输入与管理的接口。结构化查询语言语句可以嵌套,这使他具有极大的灵活性和强大的功能。

在本题中,我们要求选手实现部分比较重要和常见的 SQL 语句,并且自行设定和完成数据的存储格式与方式,并且我们不要求持久存储,选手可以自由选取存储方式。

2. 基本要求

- 1) 编程语言: C/C++ (附加库只允许标准库)

具体允许的库请参见: <http://en.cppreference.com/w/>, 但是带有 (C++11)标志是禁止的。

- 2) 运行平台: Windows Xp/vista/7/8

- 3) 编译器: VC/gcc

3. 作品提交

3.1. 提交规范

作品请以附件形式发送至大赛官方邮箱: seedcup@dian.org.cn。邮件主题命名方式: **初赛提交-队名**。

要求将完整的工程和工程文档打包提交,压缩包命名规则为: **初赛提交-你们的队名.zip**,提交的目录格式如下:

—[初赛提交-你们的队名]

├—Src

├—Bin

├—Doc

└—报名表

目录说明如下: Src 文件夹放置源代码及工程文件, Bin 文件夹放置最后生成的可执行文件, Doc 放

置说明文档。

文档中需要包含：

- 1) 程序的编译运行环境说明，包括使用的编辑器类型及版本。
- 2) 程序的设计结构及各模块的功能说明。
- 3) 对题目功能要求的完成情况，可以列举自己程序中觉得设计良好的部分，并说明为什么好。
- 4) 你认为必要的附加信息，以方便评委了解你的想法。

3.2. 提交时间

比赛终止时间为 2012/10/29 晚上 10.00，请各位选手注意提交时间（以邮件发送时间为准），我们不接受晚于终止时间提交的作品。

4. 评分说明

我们会依据选手提交上的文档和代码编译可执行文件，并将其作为主要评分标准，选手提交的可执行文件仅作为参考。另外，请特别注意输出的格式（参见 5 详细需求），**输出格式错误将直接被认为结果无效。**

项目	分值	评分标准
文档	10'	逻辑和结构清晰，描述和图例详细，不得超过 10 页。
代码规范	10'	编码的规范程度和代码的设计结构。
功能点	65'	功能点的实现情况，详细评分标准参见 5 详细需求。
加分项	25'	请参见 5.7 附加分数

5. 详细需求

5.1. 输入输出规范

1. 文本规范

- 1) 关键字不区分大写与小写的区别
- 2) 每条语句依据以分号结束
- 3) 列，表和库的名字都不能为定义的关键字，且不许包含空格，例如表名“create”是错误的

下面例子中间虽然有换行，但是依然算是一条正确的语句，语句依据分号而不是换行区分。

```
CrEate DaTABASe  
  
SeedCup;
```

2. 程序 IO 规范

我们将在命令行下面中如下调用选手提交的程序(eg: SeedCup.exe)：

```
SeedCup Input.txt Output.txt
```

Input.txt 包含输入的 SQL 语句；Output.txt 为执行的输出结果，格式总体依据 [CSV 规范](#)，具体要求请参照相关功能点。

例如，如下 Input.txt 输入和 Output.txt 输出对应的结果。

Input.txt	Output.txt
<pre>create database seedcup; use seedcup; create table test (Name text, Id int); show test column;</pre>	<pre>Name,Id</pre>

产生输出的只有 SELECT 和 SHOW，请严格依据格式输出，否则我们将判定为无效输出。

5.2. 字段类型

数据类型	形式
INT	32 位带符号的整数

FLOAT	32 位单精度浮点
TEXT	文本，长度不限

注意：

- 1) 文本类型只支持数字和字母，以及以下几种特殊字符：空格（" "），下划线（"_"）。
- 2) 当操作需要指定数据类型却没有指定时候，一律为 NONE。当该单元数据类型为 NONE 时候禁止任何对其中数据的操作，输出时用#表示。
- 3) 数据类型转换关系，INT 和 FLOAT 转换请参见 C 语言的相关定义。不需要考虑复杂的数字表示方法，例如 INT(0x55AA)和 FLOAT(3.7e-5)，仅考虑简单的十进制即可。
- 4) 创建的表名，库名和列名不区分大小写（即 SeedCup 和 SEEdcup 视为相同表），但是 TEXT 类型需要区分大小写。

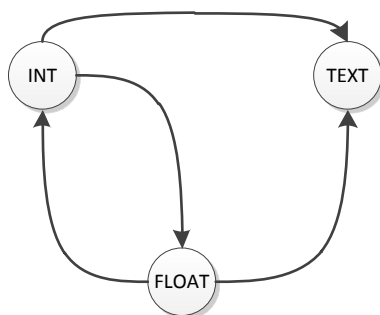


Figure 1 数据类型之间的转换关系

5.3. 操作符 (20')

操作符	描述	分数
==	等于	2
~=	不等于	2
>	大于	2
<	小于	2
>=	大于或者等于	2
<=	小于或者等于	2

BETWEEN [value1, value2]	针对 INT, FLOAT。在某个区间内部	2
LIKE [TEXT]	针对 TEXT，支持两种通配符	6
	1. '?' 匹配一个字符	
	2. '*' 匹配最多 256 个字符	
AND	必须同时满足两个条件	参见复杂的逻辑运算
OR	至少满足两个条件之间的一个	参见复杂的逻辑运算

Table 1 需要支持的操作符

- 关于 '?' 和 '*' 的详细描述请参考 <http://baike.baidu.com/view/451.htm>。
- 运算优先级，从上向下依次降低，如果运算优先级相同从左向右依次处理
 - 括号：()
 - 乘、除、求模运算符：*, /, %
 - 加减运算符：+, -
 - 比较运算符：==, ~=, >=, <=, >, <
 - 逻辑运算符：AND
 - 逻辑运算符：OR
- LIKE 和 BETWEEN 不能与其他算术符连用，例如：LIKE [abd]>0，BETWEEN [1<2]是错误的
- 除 0 和模 0 的情况视为错误处理
- ">=" 这样由多个字符组成的操作符不能被其它字符分开

5.4. 数据库定义部分(15')

1. CREATE DATABASE (1')

功能	用于创建数据库
语法	CREATE DATABASE database_name;
实例	CREATE DATABASE seedcup;

附注	数据库的名字只支持英文字母和下划线("_")
----	------------------------

2. CREATE TABLE (3')

功能	用于创建表
语法	<pre>CREATE TABLE table_name (column1 type1, column2 type2, ...);</pre>
实例	<pre>CREATE TABLE seedcup (Name TEXT);</pre>
附注	表与列的名字只支持英文字母和下划线("_") 列名不能空缺，数据类型可以空缺，最后一个逗号省略

3. ALTER TABLE (4')

功能	在已有的表中添加、修改或删除列
语法	<ol style="list-style-type: none">添加列 <pre>ALTER TABLE table_name ADD column_name datatype;</pre>删除已有列 <pre>ALTER TABLE table_name DROP COLUMN column_name;</pre>修改列的数据类型 <pre>ALTER TABLE table_name ALTER COLUMN column_name datatype;</pre>
实例	参考下面 Table 2 样表。 <ol style="list-style-type: none"><pre>ALTER TABLE Persons ADD Birthday TEXT;</pre>将在该表最后添加 Birthday 列

2. ALTER TABLE Persons ALTER COLUMN Birthday INT; 将把 Birthday 列的数据类型设置为 INT 类型
3. ALTER TABLE Person DROP COLUMN City; 将删除 City 列
- 附注
1. ADD 新加的列放在已有表的最后
 2. ADD 新添加的列不能和这个表中已有列重名
 3. 修改数据类型相关规则请参考 Figure 1
 4. 一次只能操作一列

Id	LastName	FirstName	Address	City
1	Adams	John	Oxford Street	London
2	Bush	George	Fifth Avenue	New York
3	Carter	Thomas	Changan Street	Beijing

Table 2 示例：Person 表

4. TRUNCATE TABLE (2')

- 功能 仅仅需要除去表内的数据，保留表的结构
- 语法 TRUNCATE TABLE table_name;

5. USE <DBNAME> (1')

- 功能 指定现在操作的 DB
- 语法 USE database_name;
- 附注 在对数据操作之前我们必须先 use 一个数据库，否则报错（参见错误处理）。

6. DROP TABLE/DATABASE (2')

功能	用于删除表或者数据库，可以删除当前数据库
语法	<pre>1. DROP database_name;</pre> <p>删除数据库</p> <pre>2. DROP database_name table_name;</pre> <p>删除指定数据库里面指定表</p>

7. RENAME TABLE/DATABASE (2')

功能	重命名指定表或者数据库，可以重命名当前数据库
语法	<pre>RENAME TABLE/DATABASE old_name new_name;</pre>
附注	<p>rename 不改变当前正在操作的数据库状态</p> <p>rename table 只能重命名当前正在操作的数据库中间的表</p>

5.5. 数据操作部分 (26')

我们在测试数据操作部分功能时候，只使用简单的逻辑运算表达式（单一逻辑与运算），即 where 后面只跟一个表达式。

1. SELECT (8')

功能	从表中选取数据
语法	<pre>SELECT column_name FROM table_name;</pre>
可选参数	<pre>1. WHERE 指定查询的行</pre> <pre>2. ORDER BY (DESC/INCR) 指定降序 升序排序顺序</pre>
实例	<pre>SELECT LastName, FirstName FROM Persons;</pre> <pre>SELECT * FROM Persons WHERE City=='Beijing';</pre>

输出格式	<pre>SELECT Company, OrderNumber FROM Orders ORDER BY Company DESC;</pre> <pre>SELECT * From Persons where city== 'hk' ORDER BY age DESC;</pre>
	<p>参见 Table 2 , 输入 <code>SELECT * FROM Persons WHERE City=='Beijing';</code></p> <p><i>3,Carter,Thomas,"Changan Street",Beijing</i></p> <p>每一行数据单独成行显示, 不同列的数据用逗号隔开。文本数据中有空格需要用双引号将文本包含。</p>
附注	<ol style="list-style-type: none"> 1. 星号 (*) 是选取所有列的快捷方式 2. SELECT 语句可以套嵌 (在附加分数部分测试, 不算进此项得分中) 3. ORDER BY 对文本排序依据其 ANSI 码, 没有 ORDER BY 默认按照 SELECT 输出升序排列 ; 4. 当无任何项符合条件时, 在此行输出一个"\$" 5. 有满足要求的行, 但是选择的列中间没有数据输出"#"; 输出排序中间遇到的"#", 其优先级最低

2. UPDATE (7')

功能	修改表中的数据
语法	<pre>UPDATE table_name SET (column1, column2, ...) = (new_value1, new_value2, ...)</pre> <pre>WHERE column == old_value;</pre>
实例	<p>将 Table 2 中间斜体标记'John'改为'Fred'</p> <pre>UPDATE Person SET (FirstName, City) = ('Fred', 'WuHan') WHERE LastName == 'Adams';</pre>

3. DELETE (4')

功能	删除表中的部分数据
语法	DELETE FROM table_name WHERE column == value;

4. INSERT INTO (7')

功能	向表格中插入新的行
语法	INSERT INTO table_name VALUES (value1,value2,...); 也可以指定所要插入数据的列 INSERT INTO table_name (column1,column2,...) VALUES (value1,value2,...);
实例	参见 Table 2 INSERT INTO Person VALUES ('Yao','qianming','wuhan');
附注	如果不指定列名，INSERT 括号中插入字段的数值允许为空，并且以逗号分开，如上 实例所示

5.6. 显示部分 (3')

1. SHOW DATABASES/TABLE/COLUMN (3')

功能	显示已经添加的数据库/表/列
可选参数	ORDER BY (DESC\INCR)
语法	SHOW DATABASES; SHOW database_name TABLE; SHOW table; SHOW table_name COLUMN;
说明	没有可选参数时，按照创建的顺序输出

当没有任何内容可以输出时，在此行输出一个"\$"

SHOW table 显示当前正在使用的 DATABASE 中的表，如没有正在使用的单行输出一个"\$"

输出格式 数据参见 Table 2

SHOW Person COLUMN ORDER BY DESC;

在同一行将文本全部输出

LastName,Id,FirstName,City,Address

5.7. 附加分数 (25')

1. 套嵌语句

每处需要 value 的地方都可以使用合适的 SELECT 的返回值代替。

- 给出人口多于 Russia (俄国) 的国家名称


```
SELECT name FROM world WHERE population > (SELECT population FROM bbc WHERE name=='Russia');
```
- 插入新的行，指定 Name 为在住在武汉的那一个人


```
INSERT INTO Name VALUES (SELECT Name FROM Person WHERE City == 'WuHan');
```

注意：如果嵌套的 SELECT 返回数值对于外层的 SELECT 而言，无法找到满足条件的行，则整个语句

输出"\$"。而输出"error"时，有两种典型情况：

- 套嵌的 SELECT 找不到数据，相当于外层的 SELECT 的 WHERE 条件没有赋值
- 套嵌的 SELECT 的返回值无法赋值给 WHERE 中的条件，例如 `name==(SELECT col1,col2...)`；

`name==(SELECT name FROM...)`但是套嵌 SELECT 返回多个 name。

2. 复杂的逻辑运算 (14')

在操作符部分中不计算 AND 和 OR 的分数，它们的分数在此计算。我们从简单到复杂的依次测试各种逻辑运算的情况。

```
select * from table_name Where ((( col1 == 'yao' ) or ( col2 < 24 ) and (col3 >= 2.1)) or (col4 == 'fan'))

select col1,col2 From table_name where ((( col1 == 'yao' ) or ( col2 < 24 )) and (col4 == 'fan'))
```

3. 复杂的数学运算 (5')

我们将在这里考察运算符的结合性和优先级。

```
select *from table_name where col2 < (2+4-1+(3)-2)
```

5.8. 错误处理

碰到语法错误，命令无法解析，操作不合法或者命令解析后，操作无法下发，请在输出的文件中间单行输出“error”。

一条语句仅警报一次“error”，不论错误多少次。

5.9. 测试及评分说明

- 1) 对于需要 SELECT 和 SHOW 显示输出的其它测试功能点，我们将采用简单的 SELECT 和 SHOW 语句来进行测试。
- 2) 遇到错误，程序不应该遇到错误就终止，并且应该继续向下执行命令。

5.10. 完整示例

下面这个例子将给出一个完整的操作流程。

```
create database seedcup;

use seedcup;

create table person
```

```
(  
    Id INT,  
    Name TEXT,  
    grade INT  
);
```

Id (int)	Name (text)	Grade (int)
----------	-------------	-------------

```
alter table person add address int;
```

Id (int)	Name (text)	Grade (int)	Address (int)
----------	-------------	-------------	---------------

```
alter table preson column address text;
```

Id (int)	Name (text)	Grade (int)	Address (text)
----------	-------------	-------------	----------------

```
insert into person (Id, Name, Address) values (1, 'YaoQ', 'Qiming 702');
```

Id (int)	Name (text)	Grade (int)	Address (text)
1	YaoQ		Qiming 702

```
update person set Grade=90 where Name == 'YaoQ';
```

Id (int)	Name (text)	Grade (int)	Address (text)
1	YaoQ	90	Qiming 702

```
delete from person where Id ==1;
```

Id (int)	Name (text)	Grade (int)	Address (text)
----------	-------------	-------------	----------------

```
drop seedcup person;
```

```
drop seedcup;
```

6. 参考

1. W3School 网址 : <http://www.w3school.com.cn/sql/index.asp>
2. CSV 规范 : http://en.wikipedia.org/wiki/Comma-separated_values
3. 种子杯网站 : <http://www.dian.org.cn/seedpk/?cat=9>

7. 联系我们

裁判长

姚权铭

13307124381

yaoquanming@dian.org.cn

策划	陈哲怀	15527077224	chenzhehuai@dian.org.cn
----	-----	-------------	-------------------------
