《面向对象程序设计基础》课程大作业
数据库项目实验报告

2017011620 李家昊
2016010263 董玥江
2017011550 顾清雯

2019 年 6 月 18 日

1 功能展示

本项目实现了一个类似于 MySQL 的关系型数据库管理系统，支持通过标准 SQL 语言进行数据库和表的建立，以及数据的添加、删除、查找、修改等操作。我们完成了所有的基础要求，并完成了大部分拓展需求。具体实现的功能如下：

1.1 第一阶段功能

1.1.1 Database operation

支持数据库的创建（create）、删除（drop）、切换（use）、列出（show）。

```
create database oop;
create database fop;
show databases;
/********************* Sample Output ************************
Database
fop
oop
************************************************************/
drop database fop;
show databases;
/********************* Sample Output ************************
Database
oop
************************************************************/
drop database oop;
```

1.1.2 Table management

支持表的创建（create），删除（drop），列出（show tables），详细信息获取（show columns）。

```
create database oop;
use oop;
create table aa(a double not null, primary key(a), b int not null, c char);
create table bb(a double not null, primary key(a), b int not null, c char);
create table cc(a double not null, primary key(a), b int not null, c char);
show columns from aa;
/********************* Sample Output ************************
Field   Type    Null    Key     Default Extra
a       double  NO      PRI     NULL
b       int(11) NO              NULL
c       char(1) YES             NULL
************************************************************/
show tables;
/********************* Sample Output ************************
Tables_in_oop
aa
bb
cc
************************************************************/
drop table aa;
create table dd(a double not null, primary key(a), b int not null, c char);
show tables;
/********************* Sample Output ************************
Tables_in_oop
bb
cc
dd
************************************************************/
drop database oop;
```

1.1.3 Table operation

支持数据的插入（insert）、查询（select）、更新（update）、删除（delete），以及多条件的 where clause。

```
create database oop;
use oop;
create table t(a int, b double not null, c char, primary key(b));
insert into t(a,b,c) values(1,1,'a');
insert into t(a,b,c) values(0,2,'b');
insert into t(b) values(3);
insert into t(a,b) values(3,4);
select * from t;
/********************* Sample Output ************************
a       b       c
1       1.0000  a
```

```
0       2.0000  b
NULL    3.0000  NULL
3       4.0000  NULL
************************************************************/
delete from t where c = 'a' or b > 1 and a = 0;
select * from t;
/********************** Sample Output ************************
a       b       c
NULL    3.0000  NULL
3       4.0000  NULL
************************************************************/
update t set a=4 where b=3;
select * from t;
/********************** Sample Output ************************
a       b       c
4       3.0000  NULL
3       4.0000  NULL
************************************************************/
drop database oop;
```

1.2 第二阶段基础功能

1.2.1 Load data

支持从文件中读取数据，并插入到指定表中。输入文件 1_in_file 内容如下

```
2018011343      a
2018011344      b
2018011345      c
```

请保证输入文件位于当前目录下，然后执行以下代码

```
create database oop;
use oop;
create table oop_info(stu_id int not null, stu_name char, primary key(stu_id));
load data infile '1_in_file' into table oop_info(stu_id, stu_name);
select * from oop_info;
/********************** Sample Output ************************
stu_id  stu_name
2018011343      a
2018011344      b
2018011345      c
************************************************************/
drop database oop;
```

1.2.2 Save data

支持将表中的数据导出到指定文件中。请执行以下代码。

```
create database oop;
use oop;
create table oop_info(stu_id int not null, stu_name char, primary key(stu_id));
insert into oop_info(stu_id, stu_name) values (2018011343, "a");
insert into oop_info(stu_id, stu_name) values (2018011344, "b");
insert into oop_info(stu_id, stu_name) values (2018011345, "c");
select * into outfile '2_out_file' from oop_info;
drop database oop;
```

输出文件 2_out_file 内容如下

```
2018011343      a
2018011344      b
2018011345      c
```

1.2.3 Count()

支持聚合函数 COUNT()，返回查询的记录总数。

```
create database oop;
use oop;
create table t(a char, b double, c int not null, primary key(c));
insert into t(a, b, c) values('a', 1, 1);
insert into t(a, c) values('b', 2);
insert into t(b, c) values(3, 3);
insert into t(a, c) values('d', 4);
select * from t;
/********************* Sample Output ************************
a       b       c
a       1.0000  1
b       NULL    2
NULL    3.0000  3
d       NULL    4
************************************************************/
select count(*), count(a), count(b), count(c) from t;
/********************* Sample Output ************************
COUNT(*)        COUNT(a)        COUNT(b)        COUNT(c)
4       3       2       4
************************************************************/
drop database oop;
```

1.2.4 Group by

支持分组语句 Group by，根据一个或多个列对结果集进行分组。

```
create database oop;
use oop;
create table t(a int not null, b char);
insert into t(a, b) values (1, "x");
```

```
insert into t(a, b) values (1, "x");
insert into t(a, b) values (2, "x");
insert into t(a, b) values (3, "y");
insert into t(a, b) values (3, "z");
insert into t(a, b) values (3, "z");
select b, count(*) from t group by b order by count(*);
/********************** Sample Output ************************
b       COUNT(*)
y       1
z       2
x       3
*************************************************************/
select a, b, count(*) from t group by a, b order by count(*);
/********************** Sample Output ************************
a       b       COUNT(*)
2       x       1
3       y       1
1       x       2
3       z       2
*************************************************************/
drop database oop;
```

1.2.5 Order by

支持排序语句 Order by，根据指定的表达式对结果集进行排序。

```
create database oop;
use oop;
create table t(a int, b double, primary key(b));
insert into t(a, b) values(4, 1);
insert into t(a, b) values(5, 2);
insert into t(a, b) values(3, 3);
insert into t(a, b) values(1, 4);
insert into t(a, b) values(2, 5);
select * from t order by a;
/********************** Sample Output ************************
a       b
1       4.0000
2       5.0000
3       3.0000
4       1.0000
5       2.0000
*************************************************************/
select * from t order by b;
/********************** Sample Output ************************
a       b
4       1.0000
5       2.0000
3       3.0000
1       4.0000
2       5.0000
*************************************************************/
drop database oop;
```

1.3 第二阶段拓展功能

1.3.1 Multiple tables

支持多表查询。

```
create database oop;
use oop;
create table t(a int, b int, primary key(a));
insert into t(a,b) values(-2,3);
insert into t(a,b) values(1,-1);
insert into t(a,b) values(3,0);
create table s(a int, b int, primary key(a));
insert into s(a,b) values(0,-2);
insert into s(a,b) values(1,1);
insert into s(a,b) values(-2,0);
select * from t, s order by t.a;
/********************* Sample Output ************************
t.a     t.b     s.a     s.b
-2      3       0       -2
-2      3       1       1
-2      3       -2      0
1       -1      0       -2
1       -1      1       1
1       -1      -2      0
3       0       0       -2
3       0       1       1
3       0       -2      0
************************************************************/
select * from t, s where t.a = s.a order by t.a;
/********************* Sample Output ************************
t.a     t.b     s.a     s.b
-2      3       -2      0
1       -1      1       1
************************************************************/
drop database oop;
```

1.3.2 Union

支持 Union 及 Union All 语句，支持多个 Union / Union All 语句同时使用，按左结合方式进行处理。

```
create database oop;
use oop;
create table t(a int, b double);
insert into t(a,b) values(0, 1);
insert into t(a,b) values(1, 0);
create table s(a int, b double);
insert into s(a,b) values(1, 0);
insert into s(a,b) values(2, 3);
create table r(a int, b double);
insert into r(a,b) values(2, 3);
```

```
insert into r(a,b) values(3, 5);
select a, b from t union all select a, b from s order by b;
/********************** Sample Output ************************
a       b
1       0.0000
1       0.0000
0       1.0000
2       3.0000
*************************************************************/
select a, b from t union select a, b from s order by b;
/********************** Sample Output ************************
a       b
1       0.0000
0       1.0000
2       3.0000
*************************************************************/
select a from t union all select a from s union select a from r order by a;
/********************** Sample Output ************************
a

0
1
2
3
*************************************************************/
select a from t union select a from s union all select a from r order by a;
/********************** Sample Output ************************
a

0
1
2
2
3
*************************************************************/
drop database oop;
```

1.3.3 Mathematical Functions

支持以下数字函数：abs(x), exp(x), ln(x), log10(x), ceil(x), floor(x), sin(x), cos(x), tan(x), asin(x), acos(x), atan(x), pi()，可在 select 语句以及 where 语句中使用。

支持以下逻辑运算符：and, or, xor, not，可在 select 语句以及 where 语句中使用。

```
create database oop;
use oop;
create table t(a double, b int);
insert into t(a, b) values(-1,  -1);
insert into t(a, b) values(-0.5, 0);
insert into t(a, b) values(0,    0);
insert into t(a, b) values(0.5,  1);
insert into t(a, b) values(1,    1);
insert into t(a, b) values(1.5,  2);
```

```
insert into t(a, b) values(2,    2);
select a, abs(a), exp(a), ln(a), log10(a), ceil(a), floor(a) from t;
/********************** Sample Output ************************
a       ABS(a)  EXP(a)  LN(a)   LOG10(a)        CEIL(a) FLOOR(a)
-1.0000 1.0000  0.3679  NULL    NULL    -1      -1
-0.5000 0.5000  0.6065  NULL    NULL    0       -1
0.0000  0.0000  1.0000  NULL    NULL    0       0
0.5000  0.5000  1.6487  -0.6931 -0.3010 1       0
1.0000  1.0000  2.7183  0.0000  0.0000  1       1
1.5000  1.5000  4.4817  0.4055  0.1761  2       1
2.0000  2.0000  7.3891  0.6931  0.3010  2       2
************************************************************/
select a, sin(a), cos(a), tan(a), asin(a), acos(a), atan(a) from t;
/********************** Sample Output ************************
a       SIN(a)  COS(a)  TAN(a)  ASIN(a) ACOS(a) ATAN(a)
-1.0000 -0.8415 0.5403  -1.5574 -1.5708 3.1416  -0.7854
-0.5000 -0.4794 0.8776  -0.5463 -0.5236 2.0944  -0.4636
0.0000  0.0000  1.0000  0.0000  0.0000  1.5708  0.0000
0.5000  0.4794  0.8776  0.5463  0.5236  1.0472  0.4636
1.0000  0.8415  0.5403  1.5574  1.5708  0.0000  0.7854
1.5000  0.9975  0.0707  14.1014 NULL    NULL    0.9828
2.0000  0.9093  -0.4161 -2.1850 NULL    NULL    1.1071
************************************************************/
select a, b, a-b/(a+sin(pi()/2))*cos(pi()) from t where a = b or (sin(pi()/2) = cos(0)
     xor log10(100) = abs(-2)) and not (exp(0) = tan(pi()/4) and ceil(1.5) != floor
    (2.4)) order by a;
/********************** Sample Output ************************
a       b       (a-((b/(a+SIN((3.1416/2))))*COS(3.1416)))
-1.0000 -1      NULL
0.0000  0       0.0000
1.0000  1       1.5000
2.0000  2       2.6667
************************************************************/
drop database oop;
```

1.3.4 Date & Time

实现日期和时间类，实现函数 ADDDATE(d,n) 以及 ADDTIME(t,n)，带有自动进位功能，兼容原有比较运算符，可参与排序。

```
create database oop;
use oop;
create table t(a date, b time);
insert into t(a, b) values("2019-03-13", "15:47:59");
insert into t(a, b) values("1979-09-21", "20:16:36");
insert into t(a, b) values("1949-10-01", "20:15:08");
insert into t(a, b) values("1919-05-04", "20:15:36");
select * from t order by a;
/********************** Sample Output ************************
a       b
1919-05-04      20:15:36
1949-10-01      20:15:08
1979-09-21      20:16:36
```

```
2019-03-13      15:47:59
************************************************************/
select * from t order by b;
/********************* Sample Output ************************
a       b
2019-03-13      15:47:59
1949-10-01      20:15:08
1919-05-04      20:15:36
1979-09-21      20:16:36
************************************************************/
select a, adddate(a, 20), b, addtime(b, 30) from t order by a;
/********************* Sample Output ************************
a       ADDDATE(a,20)   b       ADDTIME(b,30)
1919-05-04      1919-05-24      20:15:36        20:16:06
1949-10-01      1949-10-21      20:15:08        20:15:38
1979-09-21      1979-10-11      20:16:36        20:17:06
2019-03-13      2019-04-02      15:47:59        15:48:29
************************************************************/
drop database oop;
```

1.3.5 Operator + - * / %

支持以下算术运算符：加、减、乘、除、取模。

```
create database oop;
use oop;
create table t(a int, b int);
insert into t(a,b) values(3, 2);
insert into t(a,b) values(3, 0);
insert into t(a,b) values(0, 3);
select a, b, a+b, a-b, a*b, a/b, a%b from t order by b;
/********************* Sample Output ************************
a       b       (a+b)   (a-b)   (a*b)   (a/b)   (a%b)
3       0       3       3       0       NULL    NULL
3       2       5       1       6       1.5000  1
0       3       3       -3      0       0.0000  0
************************************************************/
drop database oop;
```

1.3.6 Operator not / and / or / xor

测试样例请参见 section 1.3.3。

1.3.7 Join

支持多表的连接（join），包括内连接（inner join）、左连接（left join）和右连接（right join），支持一个或多个 Left / Inner / Right Join 语句，按左结合方式进行处理。

```
create database oop;
use oop;
create table t(a int);
insert into t(a) values(0);
insert into t(a) values(1);
insert into t(a) values(2);
insert into t(a) values(3);
create table r(a int);
insert into r(a) values(1);
insert into r(a) values(2);
insert into r(a) values(3);
insert into r(a) values(4);
create table s(a int);
insert into s(a) values(2);
insert into s(a) values(3);
insert into s(a) values(4);
insert into s(a) values(5);
select * from t inner join r order by t.a;
/********************* Sample Output ************************
t.a     r.a
0       1
0       2
0       3
0       4
1       1
1       2
1       3
1       4
2       1
2       2
2       3
2       4
3       1
3       2
3       3
3       4
************************************************************/
select * from t inner join r on t.a = r.a order by t.a;
/********************* Sample Output ************************
t.a     r.a
1       1
2       2
3       3
************************************************************/
select * from t left join r on t.a = r.a order by t.a;
/********************* Sample Output ************************
t.a     r.a
0       NULL
1       1
2       2
3       3
************************************************************/
select * from t right join r on t.a = r.a order by r.a;
/********************* Sample Output ************************
t.a     r.a
1       1
```

```
2       2
3       3
NULL    4
************************************************************/
select * from t left join r on t.a = r.a inner join s on r.a = s.a order by t.a;
/********************** Sample Output ************************
t.a     r.a     s.a
2       2       2
3       3       3
************************************************************/
select * from t inner join r on t.a = r.a left join s on r.a = s.a order by t.a;
/********************** Sample Output ************************
t.a     r.a     s.a
1       1       NULL
2       2       2
3       3       3
************************************************************/
drop database oop;
```

1.3.8 Save at runtime

支持数据库的存档功能，且每次操作都修改对应的数据文件，重新启动程序后数据不会丢失。

```
create database oop;
use oop;
create table t(a int, b double, c char, d text, e date, f time, primary key(a));
insert into t(a,b,c,d,e,f) values(1,3,'a','hello␣world','2019-06-17','09:40:35');
insert into t(a,b,c,d,e,f) values(2,4,'b','goodbye␣world','2019-05-17','10:31:18');
select * from t;
/********************** Sample Output ************************
a       b       c       d       e       f
1       3.0000  a       hello world     2019-06-17      09:40:35
2       4.0000  b       goodbye world   2019-05-17      10:31:18
************************************************************/
```

然后 ctrl+C 强制退出，再次进入程序，执行以下代码

```
use oop;
select * from t;
/********************** Sample Output ************************
a       b       c       d       e       f
1       3.0000  a       hello world     2019-06-17      09:40:35
2       4.0000  b       goodbye world   2019-05-17      10:31:18
************************************************************/
drop database oop;
```

1.3.9 Like

支持使用 like 子句进行模糊搜索。

```
create database oop;
use oop;
create table t(a text);
insert into t(a) values("hello␣world");
insert into t(a) values("goodbye␣world");
insert into t(a) values("hell␣no");
insert into t(a) values("good␣morning");
insert into t(a) values("hello+-*()[]{}^.?$|\/");
select a from t where a like "hell%";
/********************* Sample Output ************************
a
hell no
hello world
hello+-*()[]{}^.?$|\/
************************************************************/
select a from t where a like "%wor%";
/********************* Sample Output ************************
a
goodbye world
hello world
************************************************************/
select a from t where a like "%+-*()[]{}^.?$|\/";
/********************* Sample Output ************************
a
hello+-*()[]{}^.?$|\/
************************************************************/
drop database oop;
```

1.4 其他有意义的改进

1.4.1 支持代码注释

支持行注释和块注释两种方式。也就是说，这份报告所有代码块中的测试样例都可以直接运行，而不用将注释去掉。当然，更简便的方式是使用我们提供的 autocheck.py 测试脚本，与这份报告相同的测试样例在 test/目录下能够找到。

```
create database oop;    -- This is a line comment.
create database         -- This is another line comment.
fop;
show
/*
This is a block comment.
I am invisible. You cannot see me.
*/
databases;
/********************* Sample Output ************************
Database
fop
oop
************************************************************/
drop database oop;
```

```
drop database fop;
```

1.4.2 支持同时更新多列

支持在一条 update 语句中同时更新多列的值，用逗号分隔。

```
create database oop;
use oop;
create table t(a int, b double not null, c char, primary key(b));
insert into t(a,b,c) values(1,1,'a');
insert into t(b) values(3);
select * from t;
/********************* Sample Output ************************
a       b       c
1       1.0000  a
NULL    3.0000  NULL
************************************************************/
update t set a=4, c='c', b=2 where b=3;
select * from t;
/********************* Sample Output ************************
a       b       c
1       1.0000  a
4       2.0000  c
************************************************************/
drop database oop;
```

1.4.3 支持同时删除多表

支持在一条语句中同时删除多表，用逗号分隔。

```
create database oop;
use oop;
create table a(a int);
create table b(a int);
create table c(a int);
create table d(a int);
show tables;
/********************* Sample Output ************************
Tables_in_oop
a
b
c
d
************************************************************/
drop table a, c, d;
show tables;
/********************* Sample Output ************************
Tables_in_oop
b
************************************************************/
drop database oop;
```

1.4.4 支持带括号的表达式

请参见 section 1.3.3

1.4.5 支持文本数据类型 text

请参见 section 1.3.9

2 项目优缺点分析

2.1 项目优点

1. 项目整体架构清晰，前后端高度分离。前端使用了词法解析器 Lexer 解析 Token，再通过 Parser 解析 Token 之间的关系，建立解析树；后端使用 Database Manager 管理多个 Database，Database 下管理多个 Table，Table 下管理多个词条。前后端通过接口连接。结构清晰，适合模块化开发，易于维护，可复用性强。
2. 功能强大。实现了绝大部分拓展功能，能够满足用户的大部分需求。
3. 有完善的异常处理机制。当遇到错误的 SQL 指令，或数据库操作失败时，程序将抛出异常，输出具体的错误信息，而不会突然崩溃。当然不排除未考虑到的特殊情况。
4. 用户体验良好。对于关键词不区分大小写，空格可以是一个或多个，一些符号附近（比如逗号、括号等）有无空格均可，提升了用户的输入体验。
5. 不存在内存泄漏。整个项目全面使用智能指针，坚决不用裸指针，从而杜绝了内存泄漏。
6. 在解析树和异常处理的设计中用到了多态，从而简化了代码，增强可读性。
7. 使用 doxygen 工具，生成了详细的说明文档和 UML，并通过 github pages 发布。文档链接为 `https://li-plus.github.io/SimpleSQL/index.html`。欢迎访问！

2.2 不足之处

1. 只写了一个基于内存的 BTree，但没能实现基于磁盘的 BTree，因此数据库后端基本不具有实用价值。
2. 由于时间紧迫，没有实现网络功能，因此只能在本地访问，数据库操作具有局限性。