

MyBase 使用手册 v1.0

MyBase 是用于教学的极简数据库实现，C++版本仅包含 4000 行代码

zhangqhn@foxmail.com

2019 年 2 月

目录

1 前言	1
2 简介	1
3 基本组成	2
3.1 数据类型	2
3.2 字符集	2
3.3 字面量	2
3.4 比较运算符	2
3.5 逻辑运算符	3
3.6 聚合函数	3
4 基本操作	4
4.1 打印帮助信息	4
4.2 清屏	4
4.3 退出	4
4.4 执行 SQL 文件	4
4.5 显示所有表	4
4.6 显示表的详细信息	4
4.7 创建表	4
4.8 删除表	5
4.9 插入数据	5
4.10 查询数据	5
4.11 删除数据	5
5 使用实例	6
5.1 创建表	6
5.2 插入数据	6
5.3 查询所有数据	6
5.4 查询各个课程的平均分数	7
5.5 查询张三的所有成绩	7
5.6 分页查询	7
5.7 使用字符串模糊匹配	8

1 前言

MyBase 的目标是为软件开发的初学者搭建一个简单的数据库实现环境，用来帮助学习者练习程序开发、理解数据库原理。

MyBase 的开发过程中参考过 `sqlite`、`leveldb` 及 `lua` 开源项目的思想或包含部分代码，在此特别声明并表示感谢。

本着共享开放的角度，MyBase 以 `GPL v3` 协议发布本项目所有源码。

联系我们

E-mail: zhangqhn@foxmail.com

微信公众号：



微信搜一搜

Q MyBase

2 简介

MyBase 是一个简单的数据库实现示例，目标是提供一个编程实践的平台。她具有以下几个特点：

- **简单** 大约包含 4000 行 C++ 代码，没有引用第三方库，没有使用晦涩的语法，初学者能在很短的时间内读完所有的代码。
- **包含丰富的内容** 如：词法分析、语法分析、数据过滤、分组聚合、封装、多态、数据存储、字符串模式匹配等等，涉及多个计算机学科内容。
- **丰富的扩展** 熟悉代码后您可以根据自己的想法修改或实现新的功能，如：主键、索引、多线程、网络编程、SQL 解析、排序 等等。

有了 MyBase 您可以将《C++程序设计》、《数据库系统原理》、《操作系统原理》、《数据结构与算法》、《编译原理》等课程内容融合在一个项目内，达到理论与实践相结合。

3 基本组成

3.1 数据类型

MyBase 支持以下几种数据类型：

类型名	长度（字节）	说明
int	8	
double	8	不可用于聚合（即：不可跟在 GROUP BY 后）
string		变长，小于 254 字节

3.2 字符集

MyBase 只支持 UTF8 字符集，所有字符串都需要转换成 UTF8 编码后存储。

3.3 字面量

- 整数

使用十进制数字书写，可以在数值前加+/-表示正/负值。

- 浮点数

使用十进制数字书写，用点(.)作为整数和小数的分割，可以在数值前加+/-表示正/负值。

- 字符串

使用单引号(')或双引号(")包含的字符序列。若使用单引号将字符序列包含时若字符序列中出现单引号，用两个连续的单引号表示一个单引号，双引号亦然。

示例：字符串 ab'cd"ef 字面量的正确书写如下： 'ab'cd"ef' 或 "ab'cd'"ef"

3.4 比较运算符

MyBase 支持下面几种运算符：

运算符	操作数	描述
=	二元	等于，可用于所有类型
<>	二元	不等于，可用于所有类型
>	二元	大于，可用于 int，double 类型
>=	二元	大于等于，可用于 int，double 类型
<	二元	小于，可用于 int，double 类型
<=	二元	小于等于，可用于 int，double 类型
like	二元	仅用于 string 类型，可以使用通配符： % 匹配一个或多个任意字符 _ 匹配一个任意字符
is not null	一元	值不为 null，可用于所有类型
is null	一元	值为 null，可用于所有类型

3.5 逻辑运算符

MyBase 只支持 与（AND）逻辑运算符。

3.6 聚合函数

聚合函数针对一组值执行计算并返回一个值，目前 GROUP BY 不支持 double 类型值聚合，并且不支持多个列进行聚合。

聚合函数	说明
AVG	平均值，支持 int，double 字段，返回值类型与字段类型一致。
COUNT	非 NULL 值总行数，支持所有字段，返回值为 int 类型。
MAX	最大值，支持 int，double 字段，返回值类型与字段类型一致。
MIN	最小值，支持 int，double 字段，返回值类型与字段类型一致。
SUM	和，支持 int，double 字段，返回值类型与字段类型一致。

4 基本操作

4.1 打印帮助信息

执行 `?` 或 `help` 打印帮助信息。如下所示：

```
D:\mybase\MyBase.exe
MyBase 版本: 1.0
MyBase > help
MyBase 版本: 1.0
CMD:
? | HELP      - 打印帮助信息.
CLS | CLEAR   - 清屏
EXIT | QUIT   - 退出
SOURCE <file> - 执行SQL文件
SQL:
SHOW TABLES
SHOW COLUMNS FROM <tab_name>
CREATE TABLE <tab_name>(<col_name> <INT|DOUBLE|STRING> [, ...])
DROP TABLE <tab_name>
INSERT INTO <tab_name>(<col_list>) VALUES(<val_list>)
SELECT <col_list> FROM <tab_name> [WHERE <condi_list>] [GROUP BY <col_name>] [LIMIT offset,cnt]
DELETE FROM <tab_name> [WHERE <condi_list>]
MyBase >
```

4.2 清屏

执行 `CLS` 或 `CLEAR` 清屏。

4.3 退出

执行 `EXIT` 或 `QUIT` 退出。

4.4 执行 SQL 文件

执行 `SOURCE` 后跟 sql 文件路径执行。使用此功能可以方便的导入数据到现有表中。

注意：sql 文件使用 UTF8 编码保存，否则字符串可能会出现乱码。

4.5 显示所有表

执行 `SHOW TABLES` 显示当前所有的表。

4.6 显示表的详细信息

执行 `SHOW TABLES <table_name>` 显示指定表的列信息。

4.7 创建表

使用 `CREATE TABLE` 语句创建表。目前支持整形(int)、浮点型(double)及字符串(string)三种数据类型。不支持主键及索引。创建表并显示表信息的示例如下：

```
D:\mybase\MyBase.exe
MyBase > CREATE TABLE test(id int, name string, val double)
MyBase >
MyBase > SHOW COLUMNS FROM test
test
-----
COL NAME |TYPE      |
-----
id       |INT       |
name     |STRING    |
val      |DOUBLE    |
-----
```

4.8 删除表

使用 DROP TABLE 语句删除表。

4.9 插入数据

使用 INSERT 语句插入数据，针对上面 test 表示例如下：

```
D:\mybase\MyBase.exe
MyBase > INSERT INTO test(id, name, val) values(1, 'frank', 86.5)
MyBase >
MyBase > SELECT * FROM test
-----
id      |name      |val      |
-----
1|frank    |86.5|
-----
```

4.10 查询数据

使用 SELECT 语句查询数据，可以使用 WHERE 子句对数据进行筛选，筛选

- **WHERE 子句**

支持的过滤条件见 [3.4 比较运算符](#)，多个过滤条件只能使用 AND 连接。

- **GROUP BY 子句**

在 SELECT 语句中 GROUP BY 子句能根据相同的数据进行分组，不能根据 double 类型数据进行分组，GROUP BY 子句放在 WHERE 子句之后，LIMIT 子句之前。不支持根据多个列进行聚合。

- **LIMIT 子句**

在 SELECT 语句中 LIMIT 子句用于限制返回的数据数量。

限制返回的数据量：LIMIT [cnt]

限制返回的数据量并设置 offset: LIMIT [offset], [cnt]

4.11 删除数据

使用 DELETE 语句删除指定表中的数据，可以带 WHERE 子句来删除选定行，若不指定 WHERE 则删除所有的数据。

5 使用实例

5.1 创建表

执行如下建表语句：

```
CREATE TABLE scoreinfo (id int, name string, subject string, score double)
```

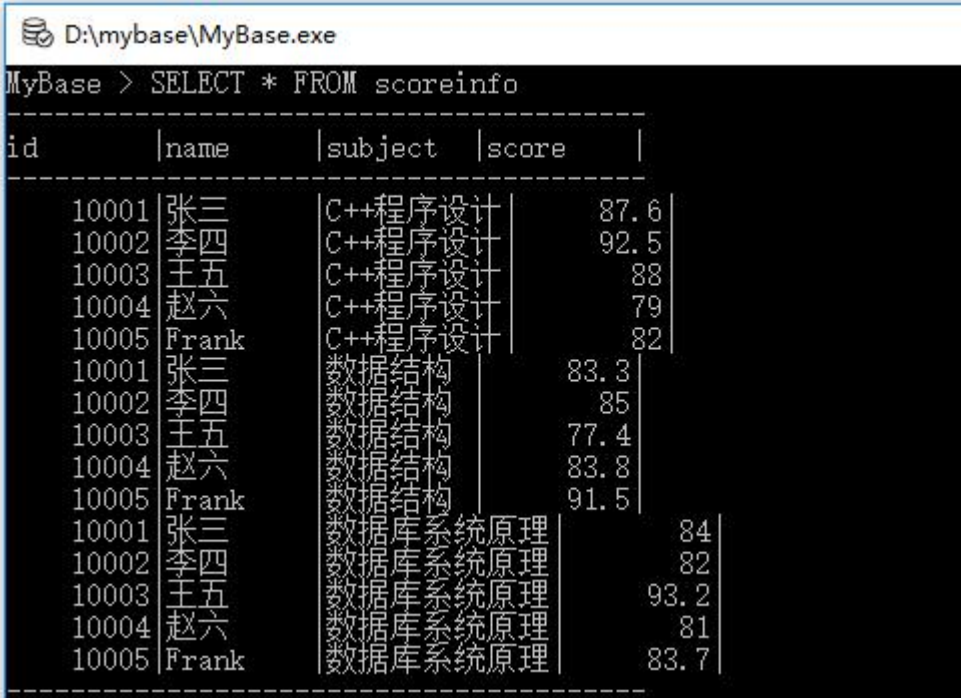
5.2 插入数据

将测试数据 score.sql（test 目录下） 使用下面的命令导入：

```
SOURCE <score.sql 的全路径>
```

注意：sql 文件使用 UTF8 编码保存，否则字符串可能会出现乱码。

5.3 查询所有数据



The screenshot shows a terminal window titled "D:\mybase\MyBase.exe". The prompt is "MyBase >". The user has entered the command "SELECT * FROM scoreinfo". The output is a table with 4 columns: id, name, subject, and score. The data is as follows:

id	name	subject	score
10001	张三	C++程序设计	87.6
10002	李四	C++程序设计	92.5
10003	王五	C++程序设计	88
10004	赵六	C++程序设计	79
10005	Frank	C++程序设计	82
10001	张三	数据结构	83.3
10002	李四	数据结构	85
10003	王五	数据结构	77.4
10004	赵六	数据结构	83.8
10005	Frank	数据结构	91.5
10001	张三	数据库系统原理	84
10002	李四	数据库系统原理	82
10003	王五	数据库系统原理	93.2
10004	赵六	数据库系统原理	81
10005	Frank	数据库系统原理	83.7

5.4 查询各个课程的平均分

```

D:\mybase\MyBase.exe
MyBase > SELECT subject, AVG(score) FROM scoreinfo GROUP BY subject
-----
subject |avg(score)|
-----
C++程序设计|    85.82|
数据结构 |    84.2|
数据库系统原理|    84.78|
-----

```

5.5 查询张三的所有成绩

```

D:\mybase\MyBase.exe
MyBase > select * from scoreinfo where name = '张三'
-----
id      |name   |subject |score  |
-----
10001   |张三   |C++程序设计|    87.6|
10001   |张三   |数据结构 |    83.3|
10001   |张三   |数据库系统原理|    84|
-----

```

5.6 分页查询

```

D:\mybase\MyBase.exe
MyBase > SELECT * FROM scoreinfo
-----
id      |name   |subject |score  |
-----
10001   |张三   |C++程序设计|    87.6|
10002   |李四   |C++程序设计|    92.5|
10003   |王五   |C++程序设计|    88|
10004   |赵六   |C++程序设计|    79|
10005   |Frank  |C++程序设计|    82|
10001   |张三   |数据结构 |    83.3|
10002   |李四   |数据结构 |    85|
10003   |王五   |数据结构 |    77.4|
10004   |赵六   |数据结构 |    83.8|
10005   |Frank  |数据结构 |    91.5|
10001   |张三   |数据库系统原理|    84|
10002   |李四   |数据库系统原理|    82|
10003   |王五   |数据库系统原理|    93.2|
10004   |赵六   |数据库系统原理|    81|
10005   |Frank  |数据库系统原理|    83.7|
-----

MyBase > SELECT * FROM scoreinfo LIMIT 2,2
-----
id      |name   |subject |score  |
-----
10003   |王五   |C++程序设计|    88|
10004   |赵六   |C++程序设计|    79|
-----

```

5.7 使用字符串模糊匹配

D:\mybase\MyBase.exe

MyBase > SELECT * FROM scoreinfo WHERE subject LIKE '数据%'

id	name	subject	score
10001	张三	数据结构	83.3
10002	李四	数据结构	85
10003	王五	数据结构	77.4
10004	赵六	数据结构	83.8
10005	Frank	数据结构	91.5
10001	张三	数据库系统原理	84
10002	李四	数据库系统原理	82
10003	王五	数据库系统原理	93.2
10004	赵六	数据库系统原理	81
10005	Frank	数据库系统原理	83.7