一、实验目的

- 设计并实现一个精简型单用户SQL引擎（DBSM）MiniSQL
- 允许用户通过字符界面输入SQL语句实现表的建立/删除；索引的建立/删除以及表记录的插入/删除/查找
- 通过对MiniSQL的设计与实现，提高学生的系统编程能力，加深对数据库系统原理的理解

二、系统需求

1. 数据类型：支持int，float，char(n)
2. 表定义：一个表最多定义32个属性，各属性可以指定是否为unique；支持unique属性的主键定义
3. 索引的建立和删除：对于表的主键自动建立B+树索引，对于声明为unique的属性可以由用户指定建立或删除索引
4. 查找记录：可以通过指定用and连接的多个条件进行查询，支持等值查询和区间查询
5. 插入和删除记录：支持每次一条记录的插入；支持每次一条或多条记录的删除
6. 语句格式与MySQL一致

三、实验环境

1. Python version>=3.7
2. Python Package: PLY, struct...

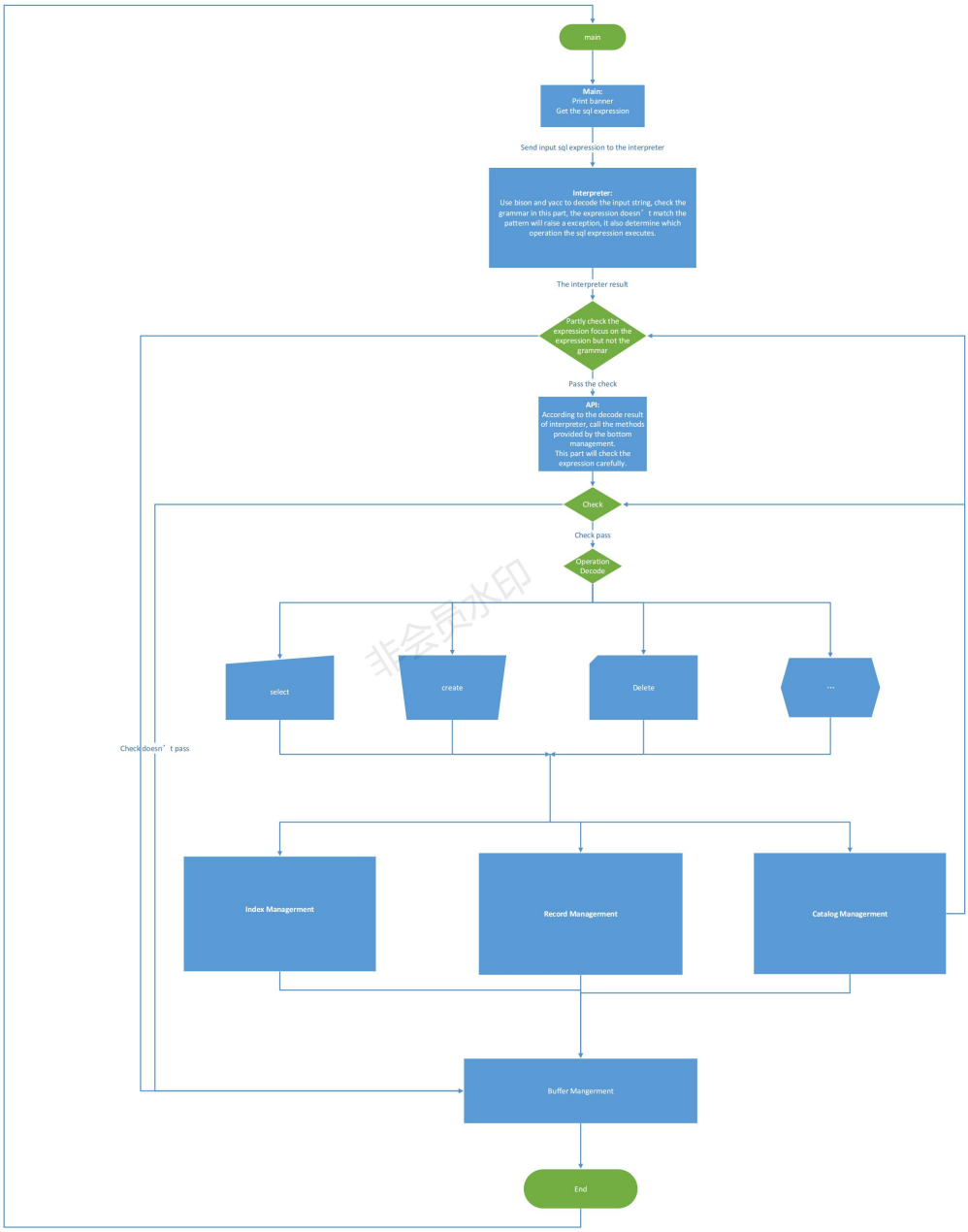
> 可以使用`pip3 install ply`安装python-lex-yacc package

四、模块设计

4.1 分工情况

- xxx：interpreter，record manager，整体调试
- xxx：部分api，buffer manager，整体调试
- xxx：部分api，index manager，catalog manager，整体调试

4.2 系统整体设计逻辑层次框架图



4.3 Interpreter

4.3.1 模块介绍

该模块使用了`python-lex-yacc` package，PLY时`lex`和`yacc`的python实现，包含了两者的大部分特性。PLY采用COC（Convention Over Configuration，惯例优于配置）的方式实现各种配置的组织。

4.3.2 词法分析

`tokens`元组定义了词法分析器可以产生的所有词法单元的类型，并且在语法分析时这个元组同样会用到，来识别终结符。

`interpreter`中共定义了如下的token list

```
tokens = (
    'LFPARENTH',
    'RGPARENTH',
    'TABLE',
    'CREATE',
    'INSERT',
```

```
    'UPDATE',
    'INTO',
    'VALUES',
    'SELECT',
    'COLUMN',
    "COMMA",
    'WHERE',
    'FROM',
    'AND',
    'SET',
    'EQUAL',
    'STAR',
    "END",
    "OP",
    "TYPE",
    "EXIT",
    "PRIMARY",
    "UNIQUE",
    "CHAR",
    "KEY",
    "DROP",
    "DELETE",
    "INDEX",
    "ON",
    "EXECFILE",
    "HELP",
    "SHOW",
    "IMPORT",
    "EXPORT"
)
```

上述`tokens`元组就包括了所有的词法单元，必须采用`tokens`作为元组的变量名，这是`PLY`强制规定的。

4.3.3 词法单元模式定义

各个词法单元的模式描述，可以采用正则表达式字符串或者函数来定义，但是必须采用t_TOKENNAME的模式命名。

词法单元规则的添加顺序：

1. 所有通过函数定义的token的添加顺序与其在词法定义文件中出现的顺序相同。
2. 所有通过正则表达式定义的token，它们按照正则表达式字符串的长度由大至小排序。

通过上面定义的顺序可以很好的处理左递归的文法。

下述是各个token与其词法单元模式定义

```
t_LFPARENTH = r'\('
t_RGPARENTH = r'\)'
t_SELECT = r'SELECT|select'
t_CREATE = r'CREATE|create'
t_INSERT = r'INSERT|insert'
t_UPDATE = r'UPDATE|update'
t_INTO = r'INTO|into'
```

```
t_VALUES = r'VALUES|values'
t_WHERE = r'WHERE|where'
t_FROM = r'FROM|from'
t_AND = r'AND|and'
t_SET = r'SET|set'
t_EQUAL = r'='
t_TABLE = r'TABLE|table'
t_COMMA = r','
t_STAR = r'\*'
t_END = r';'
t_OP = r'<>|>=|<=|<|>'
t_TYPE = r'INT|FLOAT|int|float'
t_CHAR = r'CHAR|char'
t_EXIT = r'QUIT|quit|EXIT|exit'
t_PRIMARY = r'primary|PRIMARY'
t_KEY = r'KEY|key'
t_UNIQUE = r'UNIQUE|unique'
t_DROP = r'DROP|drop'
t_DELETE = r'DELETE|delete'
t_INDEX = r'index|INDEX'
t_ON = r'ON|on'
t_EXECFILE = r'EXECFILE|execfile'
t_HELP = r'HELP|help'
t_SHOW = r'SHOW|show'
t_IMPORT = r'import|IMPORT'
t_EXPORT = r'export|EXPORT'
def t_COLUMN(t):
    r''' "[a-zA-Z0-9/ '_.-]+"|'[a-zA-Z0-9/ "_.-]+'|[a-zA-Z0-9/_.-]+ '''
    if t.value in ['FROM', 'from']:
        t.type = 'FROM'
    if t.value in ['CREATE', 'create']:
        t.type = 'CREATE'
    if t.value in ['TABLE', 'table']:
        t.type = 'TABLE'
    if t.value in ['INSERT', 'insert']:
        t.type = 'INSERT'
    if t.value in ['INTO', 'into']:
        t.type = 'INTO'
    if t.value in ['VALUES', 'values']:
        t.type = 'VALUES'
    if t.value in ['UPDATE', 'update']:
        t.type = 'UPDATE'
    if t.value in ['SET', 'set']:
        t.type = 'SET'
    if t.value in ['WHERE', 'where']:
        t.type = 'WHERE'
    if t.value in ['SELECT', 'select']:
        t.type = 'SELECT'
    if t.value in ['AND', 'and']:
        t.type = 'AND'
    if t.value in ['INT', 'int', 'FLOAT', 'float']:
        t.type = 'TYPE'
    if t.value in ['char', 'CHAR']:
        t.type = 'CHAR'
    if t.value in ['QUIT', 'quit', 'EXIT', 'exit']:
```

```
        t.type = 'EXIT'
    if t.value in ['PRIMARY', 'primary']:
        t.type = 'PRIMARY'
    if t.value in ['KEY', 'key']:
        t.type = 'KEY'
    if t.value in ['UNIQUE', 'unique']:
        t.type = 'UNIQUE'
    if t.value in ['DROP', 'drop']:
        t.type = 'DROP'
    if t.value in ['DELETE', 'delete']:
        t.type = 'DELETE'
    if t.value in ['ON', 'on']:
        t.type = 'ON'
    if t.value in ['index', 'INDEX']:
        t.type = 'INDEX'
    if t.value in ['EXECFILE', 'execfile']:
        t.type = 'EXECFILE'
    if t.value in ['HELP', 'help']:
        t.type = 'HELP'
    if t.value in ['SHOW', 'show']:
        t.type = 'SHOW'
    if t.value in ['import', 'IMPORT']:
        t.type = 'IMPORT'
    if t.value in ['export', 'EXPORT']:
        t.type = 'EXPORT'
    return t
```

其中t_column匹配除token以外的关键字的单个词法单元

4.3.4 SQL非法字符处理

```
def t_error(t):
    print("Illegal character {0}".format(t.value[0]))
    t.lexer.skip(1)
```

当检测到非法字符时，`t_error()`用于处理词法错误。这种情况下`t.value`包含了所有没有处理的输入字符串。通过`t.lexer.skip(1)`跳过1个字符，然后继续解析。

4.3.5 文法分析及相关的数据结构

各种操作的类

```
class Select(object)
class Insert(object)
class Delete(object)
class Create(object)
```

每个类中均包括一个`action()`方法，当文法解析结束，匹配对应的文法，会设置全局变量`current_action`为上述类中的一个，并调用`action`方法。

每个类中的action方法都会与`api`中的对应函数进行交互。

以及一个自己实现的堆数据结构`class stack(object)`

以下是interpreter主要定义的文法规则

```
expressions : expression
            | expressions expression
            | exp_exit
```

该文法是整棵语法分析树的根节点，因此在该文法中会调用全局变量`current_action`的`action`方法，同时如果执行过程中产生异常则使用`try except`捕获清楚`current_action`并`raise exception.`

剩下的文法及其具体实现不做详细展开，需要指出的是，在文法中也会利用catalog manager检查一部分的语法正确性，包括检查表名是否存在或不存在。如果语法正确性失败则会清楚`current_action`并`raise exception`至顶层

```
expression :  exp_select
           | exp_create_table
           | exp_create_index
           | exp_insert
           | exp_drop_table
           | exp_drop_index
           | exp_delete
           | exp_execfile
           | exp_help
           | exp_show_table
           | exp_show_index
           | exp_show
           | exp_update
           | exp_import
           | exp_export
exp_exit    : EXIT END
exp_update  : UPDATE COLUMN SET exp_assign END
            | UPDATE COLUMN SET exp_assign WHERE exp_condition END
exp_drop_table : DROP TABLE COLUMN END
exp_assign   : COLUMN EQUAL COLUMN
             | COLUMN EQUAL COLUMN COMMA exp_assign
exp_drop_index : DROP INDEX COLUMN END
exp_delete      : DELETE  FROM COLUMN END
                | DELETE  FROM COLUMN WHERE exp_condition END
exp_select : SELECT columns FROM COLUMN END
           | SELECT STAR FROM COLUMN END
           | SELECT STAR FROM COLUMN WHERE exp_condition END
           | SELECT columns FROM COLUMN WHERE exp_condition END
exp_create_table : CREATE TABLE COLUMN LFPARENTH exp_attributes COMMA
PRIMARY KEY LFPARENTH COLUMN RGPARENTH RGPARENTH END
exp_create_index : CREATE INDEX COLUMN ON COLUMN LFPARENTH COLUMN RGPARENTH
                 END
exp_attributes : exp_attribute
               | exp_attributes COMMA exp_attribute
exp_attribute : COLUMN TYPE
              | COLUMN CHAR LFPARENTH COLUMN RGPARENTH
              | COLUMN TYPE UNIQUE
              | COLUMN CHAR LFPARENTH COLUMN RGPARENTH UNIQUE
```

```
exp_insert : INSERT INTO COLUMN exp_insert_end
exp_import : IMPORT COLUMN FROM COLUMN LFPARENTH exp_attributes COMMA
PRIMARY KEY LFPARENTH COLUMN RGPARENTH RGPARENTH END
exp_export : EXPORT COLUMN FROM COLUMN END
…
```

4.3.6 解析错误处理

规则p_error(p)用来捕捉语法错误。

如果`yacc`在文法说明书中监测到错误，`yacc`会产生诊断信息并可能抛出异常，可以检测到的有：

1. Duplicated function names (if more than one rule function have the same name in the grammar file).
2. Shift/reduce and reduce/reduce conflicts generated by ambiguous grammars.
3. Badly specified grammar rules.
4. Infinite recursion (rules that can never terminate).
5. Unused rules and tokens
6. Undefined rules and tokens

4.3.7 与顶层模块交互处理

对外提供接口`interpreter(data)`进行对于输入的解析

`interpreter`方法会对输入的大小写做一定的处理后使用`yacc`对输入继续解析

需要注意的是真正的顶层实际有两个，一个是`main`模块里对`input()`函数的输入调用`interpreter`进行解析，以及自己实现的`execfile`命令对于文件输入的解析，真正的异常也都是最终`raise`至这两个模块中，异常内容由这两个模块进行输出到标准输出中。

以及提供`set_catalog(), set_api()`方法用来设置全局的`api`以及`catalog`

4.4 API

4.2.1 接口设计与说明

```
def create_table(self, tbl_name, tbl_pky, tbl_attributes)
        '''
        功能说明：
            用于创建表
        参数说明：
            tbl_name:    表名
            tbl_pky:     表的主键
            tbl_attributes: 表的属性列表
        返回值说明：
            无返回值，向终端输出信息
        '''
```

```
def drop_table(self, tbl_name)
        '''
        功能说明：
            用于删除表
        参数说明：
            tbl_name:    表名
        返回值说明：
            无返回值，向终端输出信息
        '''
```

```
def create_index(self, idx_name, idx_tbl, idx_key)
        '''
        功能说明：
            用于创建索引
        参数说明：
            idx_name:    索引名
            idx_tbl:     该索引所基于的表
            idx_key:     该索引所基于的键
        返回值说明：
            无返回值，向终端输出信息
        '''
```

```
def drop_index(self, index_name)
        '''
        功能说明：
            用于删除索引
        参数说明：
            index_name: 索引名
        返回值说明：
            无返回值，向终端输出信息
        '''
```

```
def insert_record(self, table, value, attr=None, import_flag = False)
        '''
        功能说明：
            用于在指定的表中插入记录
        参数说明：
            table:  表名
            value:  各属性的值
            attr:   指定的属性名
            import_flag: 是否来自import的调用
        返回值说明：
            无返回值，向终端输出信息
        '''
```

```
def delete_record(self, table, conditions, from_update = False)
        '''
        功能说明：
            用于在指定的表中根据条件删除对应的记录
        参数说明：
            table:  表名
            conditions: 限定条件数组
            from_update: 是否是来自于update的调用，用于确定是否输出信息
        返回值说明：
            无返回值，向终端输出信息
        '''
```

```
def select(self, table, cols, conditions)
    '''
    功能说明：
        用于在指定的表中根据条件查找对应的记录
    参数说明：
        table:  表名
        cols:   投影属性数组
        conditions: 限定条件数组
    返回值说明：
        无返回值，向终端输出信息
    '''
```

```
def update(self, table, conditions, fields)
    '''
    功能说明：
        用于在指定的表中根据条件更新对应的值
    参数说明：
        table:  表名
        conditions: 限定条件数组
        fields: 修改的属性与新的值
    返回值说明：
        无返回值，向终端输出信息
    '''
```

```
def show_index(self, index)
    '''
    功能说明：
        输出指定的索引的信息
    参数说明：
        index:  索引名
    返回值说明：
        无返回值，向终端输出信息
    '''
```

```
def show_table(self, table)
    '''
    功能说明：
        输出指定的表的信息
    参数说明：
        table:  表名
    返回值说明：
        无返回值，向终端输出信息
    '''
```

```
def output(self, table, file_path)
    '''
    功能说明：
        向指定的文件路径输出table信息，格式为csv
    参数说明：
        table:  表名
        file_path:  文件路径
    返回值说明：
        无返回值，向终端输出信息
    '''
```

4.2.2 功能描述

整个系统的核心，主要功能为提供执行SQL语句的忌口，供`Interpreter`层调用。该接口以`Interpreter`层解释生成的命令内部表示为输入，根据`Catalog Manager`提供的信息确定执行规则，并调用`Record Manager`，`Index Manager`和`Catalog Manager`提供的相应接口进行执行，最后返回执行结果给`Interpreter`模块。

4.2.3 主要数据结构

```
class optimizer():
    '''
    用于实现SQL的优化查询
    成员变量：
        catalog:    即管理元数据的catalog
    成员函数：
        check_opt:  根据输入的数据确定能否利用索引进行查询优化
    '''
```

```
class API():
    '''
    用于实现上下层的对接。这里只展示了部分API的成员函数。
    成员变量:
        catalog:    管理元数据的catalog manager
        buffer:     进行I/O操作的buffer manager
        record:     管理记录的record manager
        index:      管理索引的index manager
        optimizer:  优化用的optimizer实例
    成员函数:
        delete_record:  用于在指定的表中根据条件删除对应的记录
        select:         用于在指定的表中根据条件查询对应的记录
        update:         用于在指定的表中根据条件更新对应的值
```

```
        show_index:     输出指定的索引的信息
        show_table:     输出指定的索引的信息
        output:          向指定的文件路径输出table信息，格式为csv
    '''
```

4.5 Catalog Manager

4.5.1 模块介绍

该模块负责**管理表的目录和索引的目录**，具体包括:

1. 数据库中所有表的定义信息，包括表的名称、表中字段（列）数、主键、定义在该表上的索引。
2. 表中每个字段的定义信息，包括字段类型、是否唯一等。
3. 数据库中所有索引的定义，包括所属表、索引建立在那个字段上等。

4.5.2 数据结构与整体设计

为了方便管理，首先设计一个表示列的对象，包括名字、类型等属性，如下:

```
class Attribute():
    def __init__(self, name, type='20s', length=20, uniqueness=False):
        self.name = name
        self.uniqueness = uniqueness
        self.type = type
        self.length = length
```

其次再设计一个表示表的对象，其需要包括表名、主键、列等属性，如下:

```
class Table():
    def __init__(self, table_name, primary_key, number_attributes):
        self.table_name = table_name
        self.primary_key = primary_key
        self.attributes = []
        self.number_attributes = number_attributes
```

而由于索引包含的信息较少，只有索引名、键名和表名，因此不单独为其设计对象。直接采用字典结合列表的方式进行管理。

除去上述结构，剩下的就是主体的`catalog_manager`对象，该对象我设计了很多方法，其中**大多数是检查性质**的，为了方便上层模块的调用。由于数量众多，这些检查的方法并不在此全部列出。以及，考虑到目录文件存储的信息较少，因此我们在设计上，`catalog_manager`是直接管理目录文件的，并不需要经过buffer的管理。该类的部分接口如下:

```
class catalog_manager:
    def __init__(self)
    def save(self)
    def table_exists(self, name)
    def table_not_exists(self, name)
    def key_not_exists(self, tbl_name, key)
    def create_table(self, tbl_name, primary_key, attrlist)
    def create_index(self, index_name, tbl_name, key)
    def check_record_files(self, table)
    ...
```

初始化函数`__init__`需要实现：检查目录文件的完整性、从文件中读取目录以及检查各个表文件和索引文件的完整性。在读入目录信息时，`catalog_manager`维护一个以表名作为键的**字典**以及一个以索引名作为键的**字典**。

保存函数`save`需要实现：将目录写入文件保存起来。

检查函数`table_exists`读入一个表名，需要实现：检查该表名是否已经存在，如果存在则抛出异常。

检查函数`table_not_exists`读入一个表名，需要实现：检查该表名是否已经存在，如果不存在则抛出异常。其他的检查函数大同小异，在此不再赘述。

创建表函数`create_table`读入表名、主键以及各列的属性，需要实现：在catalog中加入该新建的表。创建索引的函数与其类似。

删除表函数`drop_table`读入表名，需要实现：在catalog中删除该表。删除索引的函数与其类似。

检查表文件函数`check_record_files`读入一个表名，需要实现：检查该表的表文件是否存在，如果不存在，则抛出异常。检查索引文件的函数与其类似。

4.6 Record Manager

4.6.1 功能描述

Record Manager主要负责记录文件的组织以及就记录的插入、删除、扫描等操作进行一定的处理之后与buffer manage模块进行交互。

4.6.2 文件组织

record文件以.rec为后缀名，使用 tbl_name + bid + off 来唯一索引记录文件中的位置，记录文件中每条记录按照8字节对齐，也就是每条记录的off一定是8的整倍数(但每个8个整倍数的offset不一定是一条record的开头)。这种组织方式会产生一定的内部碎片，但是会减少一定的外部碎片以及优化文件组织的便利性。每条记录的第一字节记录了valid byte用来标识该记录是否处在free-list中 即是否空闲。模块也对block管理、对齐进行了一些处理。

文件开头的八字节记录了文件组织所需要的相关信息，具体如下:

offset	length	attribute
0x00	1	the bool the valid byte
0x01	2	the header of free-list : block id
0x03	2	the header of free-list:offset, the offset need to multiple 8 to get the real offset
0x05	1	Bool indicates whether the free-list points to the end of the file
0x06	2	The length of a record which is assigned to 8 bytes

4.6.3 free-list组织

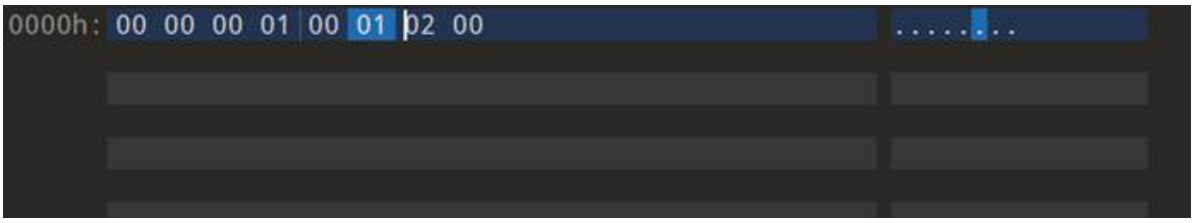
free-list构成一张单向链表，链表尾部指向最近一条被删除的记录或者当所有被删除的记录都被新记录填满时指向文件尾部。

当表被创建时，记录文件的头部被初始化成free-list指向文件末尾的(False,0,1,False,length)。

例如执行

```
create("abc", [("a", "l", 4), ("b", "l", 4)])
```

此时 abc.rec文件的结果如下

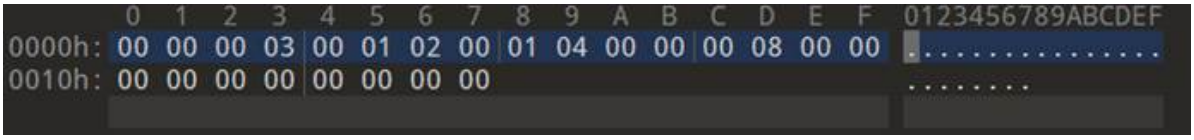


当有新记录插入时，首先取出文件头部8字节的free-list，判断此时free-list的头部是否指向文件末尾，如果此时free-list的头部指向文件末尾，则此时整个文件中不存在外部碎片，即文件内部没有空洞，则新插入的记录会插入文件尾部。如果此时free-list中的0x05 byte为false说明此时free-list的尾部指向文件中 的一个空闲的块，首先取出该空闲的块中的前8字节(该8字节指向free-list尾部的上一节点)，将该8字节存入文件头部，即将free-list的尾部取出，同时将尾部的上一节点作为尾部。同时将记录插入之前取出的free-list的尾部对应的位置。同时返回该记录插入的位置。

当文件头指向末尾时，执行插入

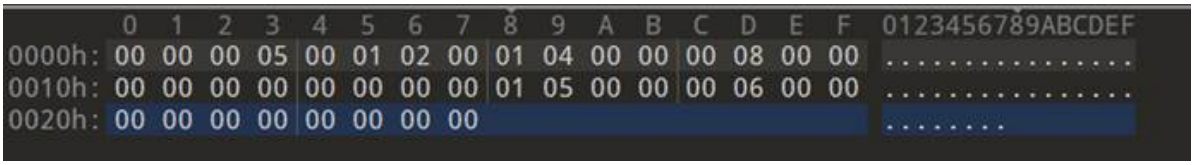
```
insert("abc", [("a", "i", 4), ("b", "i", 4)], (4, 8))
```

此时 abc.rec文件的结果如下



首先插入两条记录后删除第一条再插入第一条时

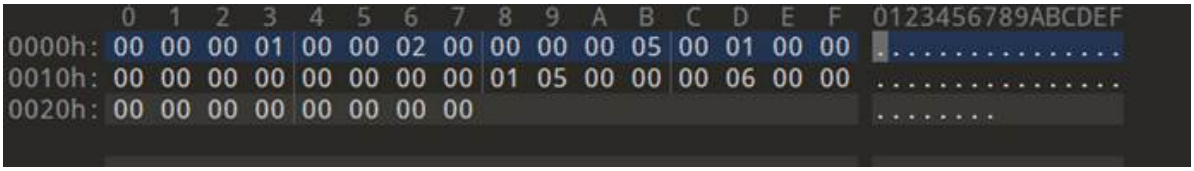
我们观察到(0,8)位置的空缺填入了第一条记录，同时文件头中free-list的尾部再次指向文件末尾。



当有记录被删除时，首先取出记录文件的头8字节即free-list的尾部，将该8字节插入被删除的记录的位置 用来构造free-list单向链表。同时将被删除的记录的信息写入记录文件的头部形成新的free-list的尾部。

首先插入两条记录后删除第一条

此时abc.rec文件中的二进制如下图所示，可以观察出文件头部free-list头部执行(0,1<<3)所在的位置，而(0,8)位置的record中的free-list节点则指向文件末尾



4.6.4 主要数据结构

该部分主要展示除free-list所涉及到的(bid,off)以外的数据结构

1. attribute

一个元组列表，用来表示一张表中各个属性的类型以及长度，元组的数据结构为(attribute name ,type, length)

其中type属性采用了python中struct的形式即

类型	type
int(4 bytes)	i
float(4 bytes)	f
char(xx bytes)	xxs

各个属性之间采用列表的形式组织在一起。利用python中的map函数与reduce函数可以方便的将各个属性按照struct的形式串联起来

```
bytes = struct.pack("=?", True)+reduce(lambda x, y: x+y, map(lambda x,y:
struct.pack(x[1], y), attr, value))
```

2. constraint

一个元组列表，用来表示扫描的条件，其中元组的数据结构为(attribute index, operation, value)

operation采用编号的形式传递

操作	编号
<	0
<=	1
>	2
>=	3
=	4
<>	5

当constraint为空时，表示将表中所有记录全部取出。

3. domain

一个元组列表，从index处获取用来，select的约束中包含已经建立索引的属性即可从index中获取对应符合约束的记录的(bid, off)的信息，(bid, off)。用来加速查找。

4.6.5 对外API列表

```
Insert(self, tbl_name, attr, value)
   """
   tbl_name        : the name of the table
   attr            : the list of the attribute tuple
   value           : the tuple of value inserted
```

```
    return value    : the tuple (bid,off)  indicates the position the new
record inserted
    """
delete_with_index(self, tbl_name, bid, off):
    """
    tbl_name        : the name of the table
    bid             : the bid of the record need to be delete
    off             : the offset of the record in the block
    return value    : none
    """
create(self, tbl_name, attr):
    """
    tbl_name        : the name of the table
    attr            : the list of the attribute tuple
    return value    : none
    """
check_record(self, record, attr, constraint):
    """
    record          : the tuple of record
    attr            : the list of the attribute tuple
    constraint      : the constraint list of tuples
    return value    : True if the record match the constraint
    """
scan_all(self, tbl_name, constraint, attr):
    """
    tbl_name        : the name of the table
    constraint      : the constraint list of tuples
    attr            : the list of the attribute tuple
    return value    : a tuple of two result list: list of the record value
and list of the position of record
    """
scan_with_index(self, tbl_name, constraint, attr, domain):
    """
    tbl_name        : the name of the table
    constraint      : the constraint list of tuples
    attr            : the list of the attribute tuple
    domain          : the list of the position of the record need to be
scanned
    return value    : a tuple of two result list: list of the record value
and list of the position of record
    """
drop_record_file(self, tbl_name):
```

4.7 Index Manager

4.7.1 模块介绍

Index Manager 负责 B+树索引的实现，实现 B+树的创建和删除（由索引的定义与删除引起）、等值查找、插入键值、删除键 值等操作，并对外提供相应的接口。 B+树中节点大小应与缓冲区的块大小相同，B+树的叉数由节点大小与索引键大小计算得到。

4.7.2 数据结构与整体设计

首先设计了一个表示节点的类，该类包含的成员有键值的列表、孩子节点的列表、保存到文件中需要用到的块的id（其实就是对B+树做一次level order之后的序号）、该节点是否为叶子结点的指示以及该节点的父母。树的order没有保存在结点中，而是保存在树的对象中，这是为了避免同样的信息被保存多次，造成信息冗余。部分接口如下：

```
class Node():
    def __init__(self, isleaf=False, parent=None):
        self.keys = []
        self.children = []
        self.bid = 0
        self.isleaf = isleaf
        self.parent = parent
    def index_of_insert(self, key)
```

此处唯一展示的方法`index_of_insert`读入一个键值，返回这个键值插入该节点后的下标。其他方法都非常简单，主要是用于判断节点处于何种状态，不叙述。

模块的主体则是名为`index_manager`的类，设计时本人尽量将与除目录外，所有与索引有关的操作都放到这里实现，比如文件的创建、读写与移除。实际编写时，考虑到编程的便捷度，没有写删除键值的方法，维护索引的方法目前只有重建索引。同时，由于直接在相应的块中维护B+树较为困难，存在着诸多如类型转换的问题，B+树索引在内存中建立，保存到文件后，搜索时不会再重建B+树。其主要接口如下：

```
class index_manager():
    def __init__(self, buffer_manager)
    def search(self, name, value, type, length)
    def search_domain(self, name, value, operate, type, length)
    def create_index(self, index_name, addresses, values, order)
    def save_Bplus(self, index_name, type, length)
    def drop_index_file(self, index_name)
```

初始化函数`__init__`读入buffer manager并保存，确保其唯一性，同时要保存B+树索引的order以及根结点。

搜索函数`search`读入索引名、键值、键的类型以及长度，需要实现：在相应的B+树索引文件中查找该键值。如果键值存在则返回其在表文件中的地址指针，否则返回`False`。

区域搜索函数`search_domain`读入索引名、键值、条件、键的类型以及长度，需要实现：在相应的B+树索引文件中查找满足条件的键值。如果存在满足条件的键值则返回它们的指针。（此函数不是由本人实现）

创建索引函数`create_index`读入索引名，order、键值列表及其地址列表，需要实现：建立相应的B+树索引。

保存索引函数`save_Bplus`读入索引名、键值的类型及其地址，需要实现：将相应的B+树索引保存到文件中。

删除索引函数`drop_index_file`读入索引名，需要实现：将该索引文件移除。

4.8 Buffer Manager

4.8.1 功能描述

本模块用于提供`MiniSQL`的I/O接口，通过模拟计算机内部的Buffer Manager与Buffer Block，以模拟缓冲区的管理。主要功能包括：

- 读取指定的数据到缓存区的一个block中；
- 修改缓存区内的数据；
- 根据LRU算法，实现缓冲区的替换算法；
- 记录缓冲区各页的状态，即是否修改以及上次访问的时间；

根据本次实验需要，设定一个block大小为4KB，缓存区共有32个block。

4.8.2 主要数据结构

```
class bufferBlock():
    '''
    每一个实例作为缓冲区的一个block
    成员变量：
        file:        内容来源的文件名
        file_bid:    内容来源在文件中的位置
        content:     数据内容
        timestamp:   上次访问的时间戳
        modified:    是否被修改过的标记
    成员函数：
        read:        按照指定的文件与位置读取指定大小的内容并返回
        read_block:  按照指定的文件与位置读取4kb内容
        write:       将修改后的内容写入缓存区
        commit:      将指定的block的内容写回磁盘文件
        refreshTimestamp:    更新时间戳
    '''
```

```
class bufferManager():
    '''
    仅在程序运行初始时形成一个实例，作为整个程序的缓存区管理者
    成员变量：
        blockArray: 32个bufferBlock实例构成的缓存区
    成员函数：
        LRU:     根据LRU算法，返回应当被替换的block的下标
        read:        按照指定的文件与位置读取指定大小的内容并返回
        read_block:  按照指定的文件与位置读取4kb内容
        write:       将修改后的内容写入缓存区
        commitOne:   将包含指定文件与位置的block写入磁盘
        commitAll:   将所有修改过的block写入磁盘
    '''
```

4.9 Database Files

在数据的二进制与实际意义转化的方式上，我们采用了`struct`库里面的`pack`与`unpack`函数，对于字符串，我们还是用了`utf-8`编码与解码。

4.9.1 Record File

即置于`/record`文件夹下面的`.rec`二进制文件，每一个表生成的时候都会创建一个对应的文件用于保存记录信息，随着表的删除而被删除。

具体的构造如下：

文件的最开始八个字节用来保存`freelist`的头结点位置信息、文件尾部标记`tail_flag`以及每条记录的长度`record_length`。

- `freelist`保存了因删除而导致的文件中的碎片空间信息，在插入是将优先占用`freelist`中保存的碎片空间。
- `tail_flag`标志当前是否为文件结尾，即标志是否为空记录文件。

其后即为记录信息，而具体的元数据信息均保存在`catalog`文件中。文件按照每`4kb`的大小划分为若干块。

每条记录根据元数据信息实现定长储存，其中`int`与`float`均保存为4字节，`char(n)`保存为n字节，内容不足的地方以`\x00`补齐。因此一旦表格的属性决定好，记录的长度也就决定了。

每块内包含整数个记录，即不存在跨块保存。

每一条记录首先记录着是否被删除，其次按照**属性顺序**依次保存数据。

删除的时候实际进行的是`lazy delete`，即修改删除标记为`True`，同时将位置加入`freelist`。

4.9.2 Index File

即置于`/index`文件下的`.ind`二进制文件，保存了索引的信息。其中储存信息的基本单位实际上是`B+`树的节点，节点长度定长。同样，文件按照每`4kb`的大小划分为若干块。每个块包含整数个节点，即也不存在跨块保存。

在每个节点的初始三个字节保存了指示**当前节点是否为叶子结点的布尔量**与**当前节点内部键值对数量的整数**。

对于非叶结点，键值对的`key`为指针指向的子节点的最小值，`value`为指向索引文件中节点位置的`(bid, offset)`坐标对。

对于叶子结点，键值对的`key`即为索引的键值，`value`为指向对应记录文件中记录所在位置的`(bid, offset)`坐标对。

储存方式采用了`level order`，这一顺序有力地方便了文件的储存读写乃至`search_domain`的实现。

4.9.3 Catalog File

即置于`/catalog`文件下的`.dat`二进制文件，保存了元数据的信息。

仅根据种类分为`index_catalog.dat`文件与`table_catalog.dat`文件。不会新增或删除文件，只会更新。

储存信息的基本单位实际上是元数据对象，长度为**不定长**。同样，文件按照每`4kb`的大小划分为若干块。每个块包含整数个节点，即也不存在跨块保存。

两个文件的最开始的4个字节都分别储存了元数据的数量，即索引的数量与表的数量。

对于`index_catalog.dat`，每个元数据对象包含了如下数据：

- 索引名字的长度`len_index`

- 表名字的长度 `len_table`
- 属性名字的长度 `len_key`
- 索引名 `index`
- 表名 `tbl`
- 属性名 `key`

对于 `table_catalog.dat`，每个元数据对象包含了如下数据：

- 表名字的长度 `len_table`
- 主键名的长度 `len_pky`
- 属性数 `num_attr`
- 表名 `table_name`
- 主键名 `primary`
- `num_attr` 个属性 `tuple`：
    - 属性名长度 `len_name`
    - 属性名 `attr_name`
    - 是否具有唯一性 `uniqueness`
    - 类型 `type`

4.9.4 CSV 文件

对于导入与导出功能，我们采用CSV文件进行储存， 并不直接指定路径。

第一行储存属性名，其后各行分别储存一行记录。

注意，我们在CSV文件中并没有保存包括属性类型等在内的元数据，导入时由用户指定，导出时根据 `catalog` 获取必要信息。

五、 系统实现分析及运行截图

在这里我们结合我们自行设计的测试代码进行运行。

Part 0 基础说明与功能概览

0.1 基础说明

1. 我们的SQL语句对大小写**不敏感**
2. 每个语句以英文分号作为结束标志
3. 我们**额外支持**的语句主要有: update, help, show, import和output，具体阐述见后

0.2 Bonus一览

1. 类似mysql的错误提示；
2. 更多语句：
    1. update;
    2. show, show table, show index;
    3. import;
    4. output;
    5. help;
3. SQL查询优化（利用index）

Part 1 基础要求语句

> 这部分全部执行正确的语句，异常检测相关语句我们单独置于Part 3进行集中测试
>
> 退出语句放在全部语句的最后进行测试
>
> index优化的测试需要有大量数据支撑才更有效果，因此放在了import语句后

1.1 创建表格

```
create table user (id int, nick_name char(10) unique, gender char(1), score
float, primary key (id));
```

此时可用`show`语句查看已经创建成功，并为主键创建了索引，且已经新建了文件：

```
show table user;
```

```
show;
```

```
sql execute file line 1>create table user (id int, nick_name char(10) unique, gender char(1), score float, primary key (id));
Successfully create table 'user'
Duration: 0.002986s

sql execute file line 2>show table user;
-----------------------------------------------------
|                   Table Infomation                |
-----------------------------------------------------
|                        user                       |
-----------------------------------------------------
|   attribute_name   |   atrribute_type   |    uniqueness    |
-----------------------------------------------------
|         id         |        int         |      primary     |
-----------------------------------------------------
|     nick_name      |      char(10)      |       True       |
-----------------------------------------------------
|       gender       |      char(1)       |       False      |
-----------------------------------------------------
|       score        |       float        |       False      |
-----------------------------------------------------

sql execute file line 3>show;
tables: dents test student2 user
indices: dents test student2 stuidx user
```

1.2 删除表格

```
drop table user;
```

成功删除表格，可用`show`语句观测到已经没有user表格及为主键创建的索引了：

```
show;
```

```
sql execute file line 4>drop table user;
Successfully drop table 'user'
Duration: 0.036899s

sql execute file line 5>show;
tables: dents test student2
indices: dents test student2 stuidx
```

为了后续的测试，在这里重建表格：

```
create table user (id int, nick_name char(10) unique, gender char(1), score
float, primary key (id));
```

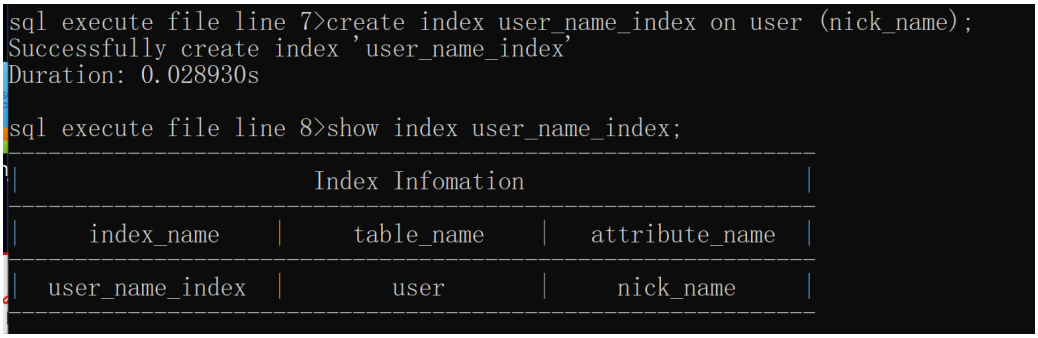
1.3 创建索引

创建表格的同时已经为主键创建了索引，这里为unique的nick_name创建索引

```
create index user_name_index on user (nick_name);
```

可用`show`语句查看

```
show index user_name_index;
```

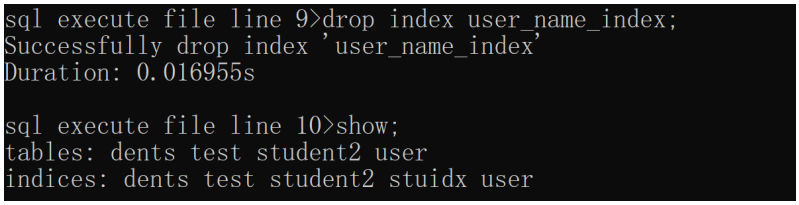


1.4 删除索引

```
drop index user_name_index;
```

可用`show`语句查看

```
show;
```



恢复索引

```
create index user_name_index on user (nick_name);
```

1.5 INSERT语句

执行以下插入测试数据的语句：

```
insert into user values (31901001, 'mike', 'M', 94.0);insert into user values
(31901002, 'john', 'M', 75.6);insert into user values (31902001, 'max', 'F',
33.3);insert into user values (31902003, 'lucy', 'F', -5.0);
```

```
sql execute file line 12>insert into user values (31901001, 'mike', 'M', 94.0);
1 row affected
Successfully insert
Duration: 0.051857s

sql execute file line 13>insert into user values (31901002, 'john', 'M', 75.6);
1 row affected
Successfully insert
Duration: 0.066790s

sql execute file line 14>insert into user values (31902001, 'max', 'F', 33.3);
1 row affected
Successfully insert
Duration: 0.072802s

sql execute file line 15>insert into user values (31902003, 'lucy', 'F', -5.0);
1 row affected
Successfully insert
Duration: 0.079347s
```

1.6 SELECT语句

1.6.1 全部属性搜索

```
select * from user;
```

```
sql execute file line 1>select * from user;
-------------------------------------------------------------------------
|        id       |    nick_name    |      gender     |      score      |
-------------------------------------------------------------------------
|     31901001    |       mike      |        m        |       94.0      |
-------------------------------------------------------------------------
|     31901002    |       john      |        m        |       75.6      |
-------------------------------------------------------------------------
|     31902001    |       max       |        f        |       33.3      |
-------------------------------------------------------------------------
|     31902003    |       lucy      |        f        |       -5.0      |
-------------------------------------------------------------------------
4 entrys in set
Duration: 0.076218s
```

1.6.2 部分属性搜索

```
select id, nick_name from user;
```

```
sql execute file line 2>select id, nick_name from user;
-----------------------------------
|        id       |    nick_name    |
-----------------------------------
|     31901001    |       mike      |
-----------------------------------
|     31901002    |       john      |
-----------------------------------
|     31902001    |       max       |
-----------------------------------
|     31902003    |       lucy      |
-----------------------------------
4 entrys in set
Duration: 0.175777s
```

1.6.3 单条件搜索

为节约测试时间起见，我们设置的测试语句有如下特点：

1. **类型不同的数据**的条件语句仅在某些有代表性的地方全部覆盖；
2. **无符合数据**的返回只在某些有代表性的地方写了相应测试语句；
3. **选择全部属性还是某几个属性**已在上述全局搜索部分做了测试，在下面的测试中我们大部分采用了全局搜索，某些地方混用了部分属性；
4. 字符串采用**大小写、单双引号**的测试也仅在相等部分做了不敏感测试。

综上，测试语句对于典型功能做了测试，由于上述的省略的功能测试与我们测试的重点功能**独立不相干**，故实无全局测试的必要，如有需要可在线下验收过程中随时提出，现场测试。

= int 相等，有符合数据&无符合数据

```
select * from user where id = 31901001;
select * from user where id = 88888888;
```

```
sql execute file line 3>select * from user where id = 31901001;
-------------------------------------------------------------------
|        id      |    nick_name    |      gender     |      score     |
-------------------------------------------------------------------
|     31901001   |       mike      |        m        |       94.0     |
-------------------------------------------------------------------
1 entrys in set
Duration: 0.151026s

sql execute file line 4>select * from user where id = 88888888;
-------------------------------------------------------------------
|        id      |    nick_name    |      gender     |      score     |
-------------------------------------------------------------------
0 entrys in set
Duration: 0.033958s
```

= char 相等

```
select * from user where nick_name = "John";
select * from user where nick_name = 'john';
```

> 大小写不敏感、单双引号不敏感

```
sql execute file line 5>select * from user where nick_name = "John";
SYNTAX Error: There is illegal variable in your SQL syntax

sql execute file line 6>select * from user where nick_name = 'john';
-------------------------------------------------------------------
|        id      |    nick_name    |      gender     |      score     |
-------------------------------------------------------------------
|     31901002   |       john      |        m        |       75.6     |
-------------------------------------------------------------------
1 entrys in set
Duration: 0.098854s
```

= float 相等

```
select * from user where score = 33.3;
```

```
sql execute file line 7>select * from user where score = 33.3;
-------------------------------------------------------------------
|        id      |    nick_name    |      gender     |      score     |
-------------------------------------------------------------------
|     31902001   |       max       |        f        |       33.3     |
-------------------------------------------------------------------
1 entrys in set
Duration: 0.160976s
```

<> int 不等

```
select id, nick_name, score from user where id <> 31901001;
select id, nick_name, score from user where id <> 88888888;
```

> 下面的语句使用的值在表中不存在

```
sql execute file line 8>select id, nick_name, score from user where id <> 31901001;
-------------------------------------------------
|        id       |    nick_name    |      score      |
-------------------------------------------------
|     31901002    |        john     |        75.6     |
-------------------------------------------------
|     31902001    |        max      |        33.3     |
-------------------------------------------------
|     31902003    |        lucy     |        -5.0     |
-------------------------------------------------
3 entrys in set
Duration: 0.182371s

sql execute file line 9>select id, nick_name, score from user where id <> 88888888;
-------------------------------------------------
|        id       |    nick_name    |      score      |
-------------------------------------------------
|     31901001    |        mike     |        94.0     |
-------------------------------------------------
|     31901002    |        john     |        75.6     |
-------------------------------------------------
|     31902001    |        max      |        33.3     |
-------------------------------------------------
|     31902003    |        lucy     |        -5.0     |
-------------------------------------------------
4 entrys in set
Duration: 0.204515s
```

<> char 不等

```
select * from user where nick_name <> 'john';
select * from user where nick_name <> 'bdz';
```

> 下面的语句使用的值在表中不存在

```
minisql>select * from user where nick_name <> 'john';
-----------------------------------------------------------------------
|        id       |    nick_name    |      gender     |      score      |
-----------------------------------------------------------------------
|     31901001    |        mike     |        m        |        94.0     |
-----------------------------------------------------------------------
|     31902001    |        max      |        f        |        33.3     |
-----------------------------------------------------------------------
|     31902003    |        lucy     |        f        |        -5.0     |
-----------------------------------------------------------------------
3 entrys in set
Duration: 0.005975s
minisql>select * from user where nick_name <> 'bdz';
-----------------------------------------------------------------------
|        id       |    nick_name    |      gender     |      score      |
-----------------------------------------------------------------------
|     31901001    |        mike     |        m        |        94.0     |
-----------------------------------------------------------------------
|     31901002    |        john     |        m        |        75.6     |
-----------------------------------------------------------------------
|     31902001    |        max      |        f        |        33.3     |
-----------------------------------------------------------------------
|     31902003    |        lucy     |        f        |        -5.0     |
-----------------------------------------------------------------------
4 entrys in set
Duration: 0.003992s
```

<> float 不等

```
select * from user where score <> -5.0;
select * from user where score <> 100;
```

> 下面的语句使用的值在表中不存在

```
sql execute file line 12>select * from user where score <> -5.0;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31901001  |       mike  |         m  |       94.0 |
+------------+-------------+------------+------------+
|  31901002  |       john  |         m  |       75.6 |
+------------+-------------+------------+------------+
|  31902001  |       max   |         f  |       33.3 |
+------------+-------------+------------+------------+
3 entrys in set
Duration: 0.210990s

sql execute file line 13>select * from user where score <> 100;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31901001  |       mike  |         m  |       94.0 |
+------------+-------------+------------+------------+
|  31901002  |       john  |         m  |       75.6 |
+------------+-------------+------------+------------+
|  31902001  |       max   |         f  |       33.3 |
+------------+-------------+------------+------------+
|  31902003  |       lucy  |         f  |       -5.0 |
+------------+-------------+------------+------------+
4 entrys in set
Duration: 0.126042s
```

> >= int

```
select * from user where id > 31901001;
select * from user where id >= 31901001;
select * from user where id > 31900000;
select * from user where id >= 31900000;
select * from user where id > 31902000;
select * from user where id >= 31902000;
select * from user where id > 31902003;
select * from user where id >= 31902003;
select * from user where id > 32000000;
select * from user where id >= 32000000;
```

> - 1, 2: 下界测试;
> - 3, 4: 数值不在表格内，下界;
> - 5, 6: 数值在表格内，中间数据;
> - 7, 8: 上界测试
> - 9, 10: 数值不在表格内，上界

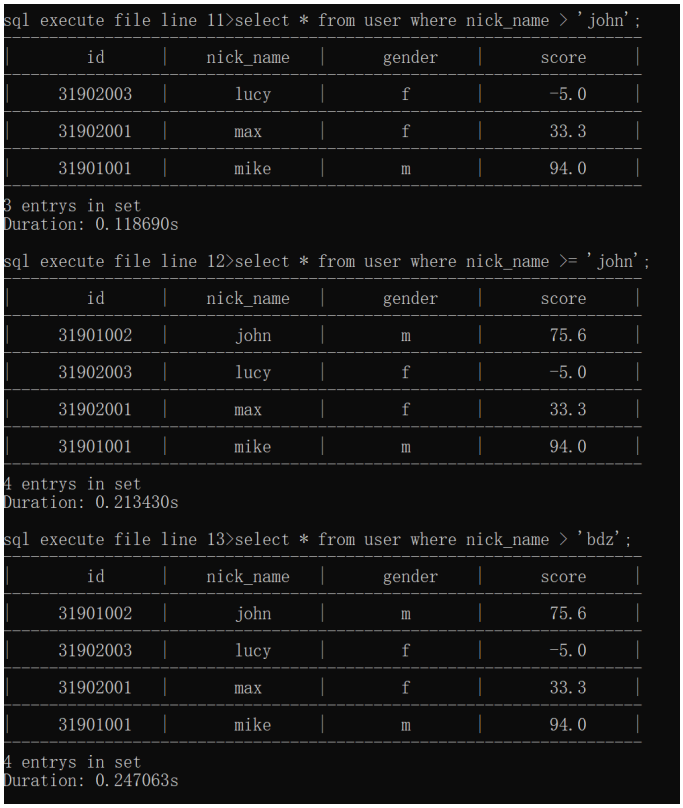
```
sql execute file line 1>select * from user where id > 31901001;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31901002  |        john |          m |       75.6 |
+------------+-------------+------------+------------+
|  31902001  |         max |          f |       33.3 |
+------------+-------------+------------+------------+
|  31902003  |        lucy |          f |       -5.0 |
+------------+-------------+------------+------------+
3 entrys in set
Duration: 0.002993s

sql execute file line 2>select * from user where id >= 31901001;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31901001  |        mike |          m |       94.0 |
+------------+-------------+------------+------------+
|  31901002  |        john |          m |       75.6 |
+------------+-------------+------------+------------+
|  31902001  |         max |          f |       33.3 |
+------------+-------------+------------+------------+
|  31902003  |        lucy |          f |       -5.0 |
+------------+-------------+------------+------------+
4 entrys in set
Duration: 0.179518s

sql execute file line 3>select * from user where id > 31900000;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31901001  |        mike |          m |       94.0 |
+------------+-------------+------------+------------+
|  31901002  |        john |          m |       75.6 |
+------------+-------------+------------+------------+
|  31902001  |         max |          f |       33.3 |
+------------+-------------+------------+------------+
|  31902003  |        lucy |          f |       -5.0 |
+------------+-------------+------------+------------+
4 entrys in set
Duration: 0.246335s
```

```
sql execute file line 4>select * from user where id >= 31900000;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31901001  |        mike |          m |       94.0 |
+------------+-------------+------------+------------+
|  31901002  |        john |          m |       75.6 |
+------------+-------------+------------+------------+
|  31902001  |         max |          f |       33.3 |
+------------+-------------+------------+------------+
|  31902003  |        lucy |          f |       -5.0 |
+------------+-------------+------------+------------+
4 entrys in set
Duration: 0.150702s

sql execute file line 5>select * from user where id > 31902000;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31902001  |         max |          f |       33.3 |
+------------+-------------+------------+------------+
|  31902003  |        lucy |          f |       -5.0 |
+------------+-------------+------------+------------+
2 entrys in set
Duration: 0.228402s

sql execute file line 6>select * from user where id >= 31902000;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31902001  |         max |          f |       33.3 |
+------------+-------------+------------+------------+
|  31902003  |        lucy |          f |       -5.0 |
+------------+-------------+------------+------------+
2 entrys in set
Duration: 0.095750s

sql execute file line 7>select * from user where id > 31902003;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
0 entrys in set
Duration: 0.049854s

sql execute file line 8>select * from user where id >= 31902003;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
|  31902003  |        lucy |          f |       -5.0 |
+------------+-------------+------------+------------+
1 entrys in set
Duration: 0.103716s

sql execute file line 9>select * from user where id > 32000000;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
0 entrys in set
Duration: 0.052861s

sql execute file line 10>select * from user where id >= 32000000;
+------------+-------------+------------+------------+
|        id  |   nick_name |     gender |      score |
+------------+-------------+------------+------------+
0 entrys in set
```

> >= char

```
select * from user where nick_name > 'john';
select * from user where nick_name >= 'john';
select * from user where nick_name > 'bdz';
select * from user where nick_name >= 'bdz';
select * from user where nick_name > 'mike';
select * from user where nick_name >= 'mike';
select * from user where nick_name > 'max';
select * from user where nick_name >= 'max';
select * from user where nick_name > 'zpq';
select * from user where nick_name >= 'zpq';
```

- 1, 2: 下界测试;
- 3, 4: 数值不在表格内，下界;
- 5, 6: 数值在表格内，中间数据;
- 7, 8: 上界测试
- 9, 10: 数值不在表格内，上界



```
sql execute file line 11>select * from user where nick_name > 'john';
+-----------+-----------+-----------+-----------+
|        id |  nick_name|     gender|      score|
+-----------+-----------+-----------+-----------+
|  31902003 |       lucy|          f|       -5.0|
+-----------+-----------+-----------+-----------+
|  31902001 |        max|          f|       33.3|
+-----------+-----------+-----------+-----------+
|  31901001 |       mike|          m|       94.0|
+-----------+-----------+-----------+-----------+
3 entrys in set
Duration: 0.118690s

sql execute file line 12>select * from user where nick_name >= 'john';
+-----------+-----------+-----------+-----------+
|        id |  nick_name|     gender|      score|
+-----------+-----------+-----------+-----------+
|  31901002 |       john|          m|       75.6|
+-----------+-----------+-----------+-----------+
|  31902003 |       lucy|          f|       -5.0|
+-----------+-----------+-----------+-----------+
|  31902001 |        max|          f|       33.3|
+-----------+-----------+-----------+-----------+
|  31901001 |       mike|          m|       94.0|
+-----------+-----------+-----------+-----------+
4 entrys in set
Duration: 0.213430s

sql execute file line 13>select * from user where nick_name > 'bdz';
+-----------+-----------+-----------+-----------+
|        id |  nick_name|     gender|      score|
+-----------+-----------+-----------+-----------+
|  31901002 |       john|          m|       75.6|
+-----------+-----------+-----------+-----------+
|  31902003 |       lucy|          f|       -5.0|
+-----------+-----------+-----------+-----------+
|  31902001 |        max|          f|       33.3|
+-----------+-----------+-----------+-----------+
|  31901001 |       mike|          m|       94.0|
+-----------+-----------+-----------+-----------+
4 entrys in set
Duration: 0.247063s
```

```
sql execute file line 14>select * from user where nick_name >= 'bdz';
+----------+-----------+--------+--------+
|    id    | nick_name | gender |  score |
+----------+-----------+--------+--------+
| 31901002 |      john |      m |   75.6 |
| 31902003 |      lucy |      f |   -5.0 |
| 31902001 |       max |      f |   33.3 |
| 31901001 |      mike |      m |   94.0 |
+----------+-----------+--------+--------+
4 entrys in set
Duration: 0.098728s

sql execute file line 15>select * from user where nick_name > 'mike';
+----+-----------+--------+-------+
| id | nick_name | gender | score |
+----+-----------+--------+-------+
+----+-----------+--------+-------+
0 entrys in set
Duration: 0.145601s

sql execute file line 16>select * from user where nick_name >= 'mike';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
| 31901001 |      mike |      m |  94.0 |
+----------+-----------+--------+-------+
1 entrys in set
Duration: 0.212706s

sql execute file line 17>select * from user where nick_name > 'max';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
| 31901001 |      mike |      m |  94.0 |
+----------+-----------+--------+-------+
1 entrys in set
Duration: 0.143613s

sql execute file line 18>select * from user where nick_name >= 'max';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
| 31902001 |       max |      f |  33.3 |
| 31901001 |      mike |      m |  94.0 |
+----------+-----------+--------+-------+
2 entrys in set
Duration: 0.213428s

sql execute file line 19>select * from user where nick_name > 'zpq';
+----+-----------+--------+-------+
| id | nick_name | gender | score |
+----+-----------+--------+-------+
+----+-----------+--------+-------+
0 entrys in set
Duration: 0.140683s

sql execute file line 20>select * from user where nick_name >= 'zpq';
+----+-----------+--------+-------+
| id | nick_name | gender | score |
+----+-----------+--------+-------+
+----+-----------+--------+-------+
0 entrys in set
Duration: 0.199458s
```

> >= float

```
select * from user where score > -5.0;
select * from user where score >= -5.0;
select * from user where score > -10.0;
select * from user where score >= -10;
select * from user where score > 75.6;
select * from user where score >= 75.6;
select * from user where score > 94.0;
select * from user where score >= 94.0;
select * from user where score > 100.5;
select * from user where score >= 100.5;
```

- 1, 2: 下界测试;
- 3, 4: 数值不在表格内，下界，4并没有显示表明小数点;
- 5, 6: 数值在表格内，中间数据;
- 7, 8: 上界测试
- 9, 10: 数值不在表格内，上界

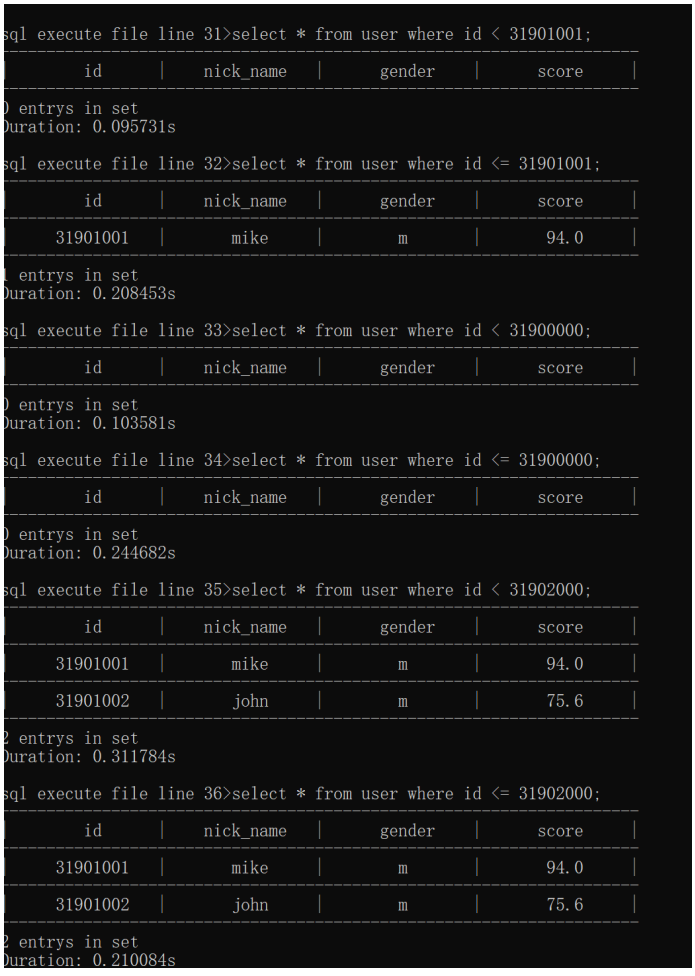
```
sql execute file line 21>select * from user where score > -5.0;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
| 31901001  |   mike    |   m    |  94.0 |
+-----------+-----------+--------+-------+
| 31901002  |   john    |   m    |  75.6 |
+-----------+-----------+--------+-------+
| 31902001  |   max     |   f    |  33.3 |
+-----------+-----------+--------+-------+
3 entrys in set
Duration: 0.031963s

sql execute file line 22>select * from user where score >= -5.0;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
| 31901001  |   mike    |   m    |  94.0 |
+-----------+-----------+--------+-------+
| 31901002  |   john    |   m    |  75.6 |
+-----------+-----------+--------+-------+
| 31902001  |   max     |   f    |  33.3 |
+-----------+-----------+--------+-------+
| 31902003  |   lucy    |   f    |  -5.0 |
+-----------+-----------+--------+-------+
4 entrys in set
Duration: 0.185520s

sql execute file line 23>select * from user where score > -10.0;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
| 31901001  |   mike    |   m    |  94.0 |
+-----------+-----------+--------+-------+
| 31901002  |   john    |   m    |  75.6 |
+-----------+-----------+--------+-------+
| 31902001  |   max     |   f    |  33.3 |
+-----------+-----------+--------+-------+
| 31902003  |   lucy    |   f    |  -5.0 |
+-----------+-----------+--------+-------+
4 entrys in set
Duration: 0.089535s
```

```
sql execute file line 24>select * from user where score >= -10;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
| 31901001  |   mike    |   m    |  94.0 |
+-----------+-----------+--------+-------+
| 31901002  |   john    |   m    |  75.6 |
+-----------+-----------+--------+-------+
| 31902001  |   max     |   f    |  33.3 |
+-----------+-----------+--------+-------+
| 31902003  |   lucy    |   f    |  -5.0 |
+-----------+-----------+--------+-------+
4 entrys in set
Duration: 0.125182s

sql execute file line 25>select * from user where score > 75.6;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
| 31901001  |   mike    |   m    |  94.0 |
+-----------+-----------+--------+-------+
1 entrys in set
Duration: 0.063770s

sql execute file line 26>select * from user where score >= 75.6;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
| 31901001  |   mike    |   m    |  94.0 |
+-----------+-----------+--------+-------+
1 entrys in set
Duration: 0.190500s

sql execute file line 27>select * from user where score > 94.0;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
0 entrys in set
Duration: 0.087767s

sql execute file line 28>select * from user where score >= 94.0;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
| 31901001  |   mike    |   m    |  94.0 |
+-----------+-----------+--------+-------+
1 entrys in set
Duration: 0.003969s

sql execute file line 29>select * from user where score > 100.5;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
0 entrys in set
Duration: 0.002988s

sql execute file line 30>select * from user where score >= 100.5;
+-----------+-----------+--------+-------+
|    id     | nick_name | gender | score |
+-----------+-----------+--------+-------+
0 entrys in set
Duration: 0.109711s
```

< <= int

```
select * from user where id < 31901001;
select * from user where id <= 31901001;
select * from user where id < 31900000;
select * from user where id <= 31900000;
select * from user where id < 31902000;
select * from user where id <= 31902000;
select * from user where id < 31902003;
select * from user where id <= 31902003;
select * from user where id < 32000000;
select * from user where id <= 32000000;
```

> - 1, 2: 下界测试;
> - 3, 4: 数值不在表格内，下界;
> - 5, 6: 数值在表格内，中间数据;
> - 7, 8: 上界测试
> - 9, 10: 数值不在表格内，上界



```
sql execute file line 37>select * from user where id < 31902003;
+-----------+-----------+--------+--------+
|    id     | nick_name | gender | score  |
+-----------+-----------+--------+--------+
| 31901001  |   mike    |   m    |  94.0  |
+-----------+-----------+--------+--------+
| 31901002  |   john    |   m    |  75.6  |
+-----------+-----------+--------+--------+
| 31902001  |   max     |   f    |  33.3  |
+-----------+-----------+--------+--------+
3 entrys in set
Duration: 0.380358s

sql execute file line 38>select * from user where id <= 31902003;
+-----------+-----------+--------+--------+
|    id     | nick_name | gender | score  |
+-----------+-----------+--------+--------+
| 31901001  |   mike    |   m    |  94.0  |
+-----------+-----------+--------+--------+
| 31901002  |   john    |   m    |  75.6  |
+-----------+-----------+--------+--------+
| 31902001  |   max     |   f    |  33.3  |
+-----------+-----------+--------+--------+
| 31902003  |   lucy    |   f    |  -5.0  |
+-----------+-----------+--------+--------+
4 entrys in set
Duration: 0.243771s

sql execute file line 39>select * from user where id < 32000000;
+-----------+-----------+--------+--------+
|    id     | nick_name | gender | score  |
+-----------+-----------+--------+--------+
| 31901001  |   mike    |   m    |  94.0  |
+-----------+-----------+--------+--------+
| 31901002  |   john    |   m    |  75.6  |
+-----------+-----------+--------+--------+
| 31902001  |   max     |   f    |  33.3  |
+-----------+-----------+--------+--------+
| 31902003  |   lucy    |   f    |  -5.0  |
+-----------+-----------+--------+--------+
4 entrys in set
Duration: 0.483521s

sql execute file line 40>select * from user where id <= 32000000;
+-----------+-----------+--------+--------+
|    id     | nick_name | gender | score  |
+-----------+-----------+--------+--------+
| 31901001  |   mike    |   m    |  94.0  |
+-----------+-----------+--------+--------+
| 31901002  |   john    |   m    |  75.6  |
+-----------+-----------+--------+--------+
| 31902001  |   max     |   f    |  33.3  |
+-----------+-----------+--------+--------+
| 31902003  |   lucy    |   f    |  -5.0  |
+-----------+-----------+--------+--------+
4 entrys in set
Duration: 0.300601s
```

< <= char

```sql
select * from user where nick_name < 'john';
select * from user where nick_name <= 'john';
select * from user where nick_name < 'bdz';
select * from user where nick_name <= 'bdz';
select * from user where nick_name < 'mike';
select * from user where nick_name <= 'mike';
select * from user where nick_name < 'max';
select * from user where nick_name <= 'max';
select * from user where nick_name < 'zpq';
select * from user where nick_name <= 'zpq';
```

> - 1, 2: 下界测试;
> - 3, 4: 数值不在表格内，下界;
> - 5, 6: 数值在表格内，中间数据;
> - 7, 8: 上界测试
> - 9, 10: 数值不在表格内，上界

```
sql execute file line 46>select * from user where nick_name <= 'mike';
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
|  31902003  |    lucy     |     f      |    -5.0    |
+------------+-------------+------------+------------+
|  31902001  |    max      |     f      |    33.3    |
+------------+-------------+------------+------------+
|  31901001  |    mike     |     m      |    94.0    |
+------------+-------------+------------+------------+
4 entrys in set
Duration: 0.328487s

sql execute file line 47>select * from user where nick_name < 'max';
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
|  31902003  |    lucy     |     f      |    -5.0    |
+------------+-------------+------------+------------+
2 entrys in set
Duration: 0.197556s

sql execute file line 48>select * from user where nick_name <= 'max';
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
|  31902003  |    lucy     |     f      |    -5.0    |
+------------+-------------+------------+------------+
|  31902001  |    max      |     f      |    33.3    |
+------------+-------------+------------+------------+
3 entrys in set
Duration: 0.313483s

sql execute file line 49>select * from user where nick_name < 'zpq';
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
|  31902003  |    lucy     |     f      |    -5.0    |
+------------+-------------+------------+------------+
|  31902001  |    max      |     f      |    33.3    |
+------------+-------------+------------+------------+
|  31901001  |    mike     |     m      |    94.0    |
+------------+-------------+------------+------------+
4 entrys in set
Duration: 0.425963s

sql execute file line 50>select * from user where nick_name <= 'zpq';
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
|  31902003  |    lucy     |     f      |    -5.0    |
+------------+-------------+------------+------------+
|  31902001  |    max      |     f      |    33.3    |
+------------+-------------+------------+------------+
|  31901001  |    mike     |     m      |    94.0    |
+------------+-------------+------------+------------+
4 entrys in set
Duration: 0.375351s
```

< <= float

```sql
select * from user where score < -5.0;
select * from user where score <= -5.0;
select * from user where score < -10.0;
select * from user where score <= -10;
select * from user where score < 75.6;
select * from user where score <= 75.6;
select * from user where score < 94.0;
select * from user where score <= 94.0;
select * from user where score < 100.5;
select * from user where score <= 100.5;
```

> - 1, 2: 下界测试;
> - 3, 4: 数值不在表格内，下界，4并没有显示表明小数点;
> - 5, 6: 数值在表格内，中间数据;
> - 7, 8: 上界测试
> - 9, 10: 数值不在表格内，上界

```
sql execute file line 41>select * from user where nick_name < 'john';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
0 entrys in set
Duration: 0.101671s

sql execute file line 42>select * from user where nick_name <= 'john';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
| 31901002 |      john |      m |  75.6 |
+----------+-----------+--------+-------+
1 entrys in set
Duration: 0.200558s

sql execute file line 43>select * from user where nick_name < 'bdz';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
0 entrys in set
Duration: 0.106185s

sql execute file line 44>select * from user where nick_name <= 'bdz';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
0 entrys in set
Duration: 0.140567s

sql execute file line 45>select * from user where nick_name < 'mike';
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
| 31901002 |      john |      m |  75.6 |
+----------+-----------+--------+-------+
| 31902003 |      lucy |      f |  -5.0 |
+----------+-----------+--------+-------+
| 31902001 |       max |      f |  33.3 |
+----------+-----------+--------+-------+
3 entrys in set
Duration: 0.205649s
```

```
sql execute file line 51>select * from user where score < -5.0;
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
0 entrys in set
Duration: 0.060510s

sql execute file line 52>select * from user where score <= -5.0;
+----------+-----------+--------+-------+
|    id    | nick_name | gender | score |
+----------+-----------+--------+-------+
| 31902003 |      lucy |      f |  -5.0 |
+----------+-----------+--------+-------+
1 entrys in set
Duration: 0.040290s
```

```
sql execute file line 53>select * from user where score < -10.0;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+

0 entrys in set
Duration: 0.151782s

sql execute file line 54>select * from user where score <= -10;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+

0 entrys in set
Duration: 0.108706s

sql execute file line 55>select * from user where score < 75.6;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+
|    31901002 |        john |           m |        75.6 |
+-------------+-------------+-------------+-------------+
|    31902001 |         max |           f |        33.3 |
+-------------+-------------+-------------+-------------+
|    31902003 |        lucy |           f |        -5.0 |
+-------------+-------------+-------------+-------------+

3 entrys in set
Duration: 0.374252s

sql execute file line 56>select * from user where score <= 75.6;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+
|    31901002 |        john |           m |        75.6 |
+-------------+-------------+-------------+-------------+
|    31902001 |         max |           f |        33.3 |
+-------------+-------------+-------------+-------------+
|    31902003 |        lucy |           f |        -5.0 |
+-------------+-------------+-------------+-------------+

3 entrys in set
```

```
sql execute file line 57>select * from user where score < 94.0;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+
|    31901002 |        john |           m |        75.6 |
+-------------+-------------+-------------+-------------+
|    31902001 |         max |           f |        33.3 |
+-------------+-------------+-------------+-------------+
|    31902003 |        lucy |           f |        -5.0 |
+-------------+-------------+-------------+-------------+

3 entrys in set
Duration: 0.305778s

sql execute file line 58>select * from user where score <= 94.0;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+
|    31901001 |        mike |           m |        94.0 |
+-------------+-------------+-------------+-------------+
|    31901002 |        john |           m |        75.6 |
+-------------+-------------+-------------+-------------+
|    31902001 |         max |           f |        33.3 |
+-------------+-------------+-------------+-------------+
|    31902003 |        lucy |           f |        -5.0 |
+-------------+-------------+-------------+-------------+

4 entrys in set
Duration: 0.308891s
```

```
sql execute file line 59>select * from user where score < 100.5;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+
|    31901001 |        mike |           m |        94.0 |
+-------------+-------------+-------------+-------------+
|    31901002 |        john |           m |        75.6 |
+-------------+-------------+-------------+-------------+
|    31902001 |         max |           f |        33.3 |
+-------------+-------------+-------------+-------------+
|    31902003 |        lucy |           f |        -5.0 |
+-------------+-------------+-------------+-------------+

4 entrys in set
Duration: 0.253286s

sql execute file line 60>select * from user where score <= 100.5;
+-------------+-------------+-------------+-------------+
|        id   |   nick_name |      gender |       score |
+-------------+-------------+-------------+-------------+
|    31901001 |        mike |           m |        94.0 |
+-------------+-------------+-------------+-------------+
|    31901002 |        john |           m |        75.6 |
+-------------+-------------+-------------+-------------+
|    31902001 |         max |           f |        33.3 |
+-------------+-------------+-------------+-------------+
|    31902003 |        lucy |           f |        -5.0 |
+-------------+-------------+-------------+-------------+

4 entrys in set
Duration: 0.311979s
```

1.6.4 多条件搜索

鉴于1.6.3做了较为全面的搜索，我们这里只进行抽样测试

同类型，双条件

```
select * from user where id < 31902002 and id >= 31901001;
```

```
sql execute file line 61>select * from user where id < 31902002 and id >= 31901001;
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901001  |    mike     |     m      |    94.0    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
|  31902001  |    max      |     f      |    33.3    |
+------------+-------------+------------+------------+
3 entrys in set
Duration: 0.208572s
```

不同类型，多条件

```
select * from user where score >= 60.0 and id <= 31902000 and gender = 'M';
```

```
sql execute file line 62>select * from user where score >= 60.0 and id <= 31902000 and gender = 'M';
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901001  |    mike     |     m      |    94.0    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
2 entrys in set
Duration: 0.172522s
```

1.7 DELETE语句

单条件， 单条删除

```
delete from user where id = 31901001;
select * from user;
```

```
sql execute file line 1>delete from user where id = 31901001;
1 entrys affected
Successfully delete
Duration: 0.002993s

sql execute file line 2>select * from user;
+------------+-------------+------------+------------+
|     id     |  nick_name  |   gender   |   score    |
+------------+-------------+------------+------------+
|  31901002  |    john     |     m      |    75.6    |
+------------+-------------+------------+------------+
|  31902001  |    max      |     f      |    33.3    |
+------------+-------------+------------+------------+
|  31902003  |    lucy     |     f      |    -5.0    |
+------------+-------------+------------+------------+
3 entrys in set
Duration: 0.001003s
```

恢复数据

```
insert into user values (31901001, 'mike', 'M', 94.0);
```

单条件，多条删除

```
delete from user where id >= 31901002;
select * from user;
```

```
sql execute file line 4>delete from user where id >= 31901002;
3 entrys affected
Successfully delete
Duration: 0.001994s

sql execute file line 5>select * from user;
+-------------+-------------+-------------+-------------+
|      id     |  nick_name  |    gender   |    score    |
+-------------+-------------+-------------+-------------+
|  31901001   |    mike     |      m      |     94.0    |
+-------------+-------------+-------------+-------------+
1 entrys in set
Duration: 0.000000s
```

恢复数据

```
insert into user values (31901002, 'john', 'M', 75.6);insert into user values
(31902001, 'max', 'F', 33.3);insert into user values (31902003, 'lucy', 'F',
-5.0);
```

多条件，单条删除

```
delete from user where id <= 31901002 and score > 80.0;select * from user;
```

```
sql execute file line 9>delete from user where id <= 31901002 and score > 80.0;
1 entrys affected
Successfully delete
Duration: 0.006981s

sql execute file line 10>select * from user;
+-------------+-------------+-------------+-------------+
|      id     |  nick_name  |    gender   |    score    |
+-------------+-------------+-------------+-------------+
|  31902003   |    lucy     |      f      |     -5.0    |
+-------------+-------------+-------------+-------------+
|  31902001   |    max      |      f      |     33.3    |
+-------------+-------------+-------------+-------------+
|  31901002   |    john     |      m      |     75.6    |
+-------------+-------------+-------------+-------------+
3 entrys in set
Duration: 0.000998s
```

恢复数据

```
insert into user values (31901001, 'mike', 'M', 94.0);
```

多条件，多条删除

```
delete from user where id > 31902000 and gender <> 'M';select * from user;
```

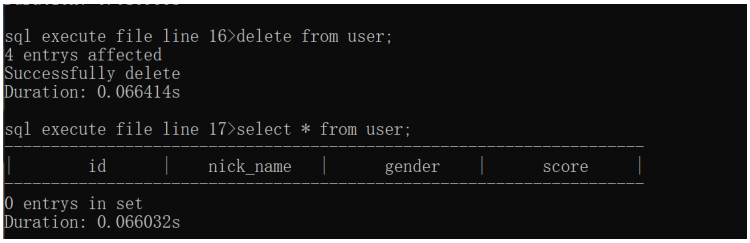
```
sql execute file line 11>insert into user values (31901001, 'mike', 'M', 94.0);
1 row affected
Successfully insert
Duration: 0.031114s

sql execute file line 12>delete from user where id > 31902000 and gender <> 'M';
2 entrys affected
Successfully delete
Duration: 0.003984s

sql execute file line 13>select * from user;
+-------------+-------------+-------------+-------------+
|      id     |  nick_name  |    gender   |    score    |
+-------------+-------------+-------------+-------------+
|  31901001   |    mike     |      m      |     94.0    |
+-------------+-------------+-------------+-------------+
|  31901002   |    john     |      m      |     75.6    |
+-------------+-------------+-------------+-------------+
2 entrys in set
Duration: 0.007978s
```

恢复数据

```
insert into user values (31902001, 'max', 'F', 33.3);insert into user values
(31902003, 'lucy', 'F', -5.0);
```

全部删除

```
delete from user;select * from user;
```



恢复数据

```
insert into user values (31901001, 'mike', 'M', 94.0);insert into user values
(31901002, 'john', 'M', 75.6);insert into user values (31902001, 'max', 'F',
33.3);insert into user values (31902003, 'lucy', 'F', -5.0);
```

Part 2 扩展功能测试

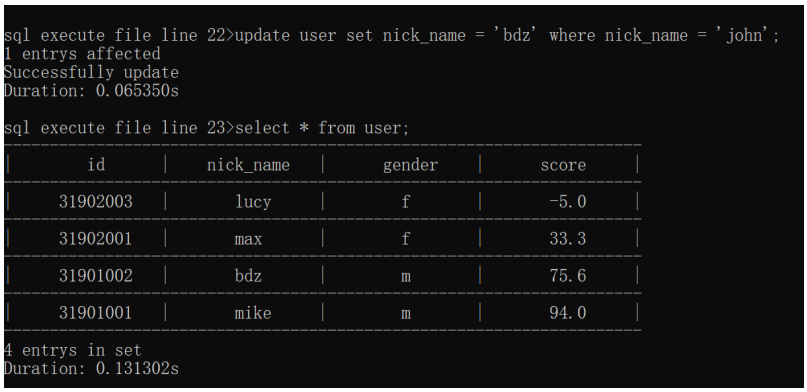
2.1 update

语句格式:

```
update 表名 set 列名1 = 值1, 列名2 = 值2, ... where 条件子句;
```

单条件, 单条单列更新

```
update user set nick_name = 'bdz' where nick_name = 'john';select * from user;
```

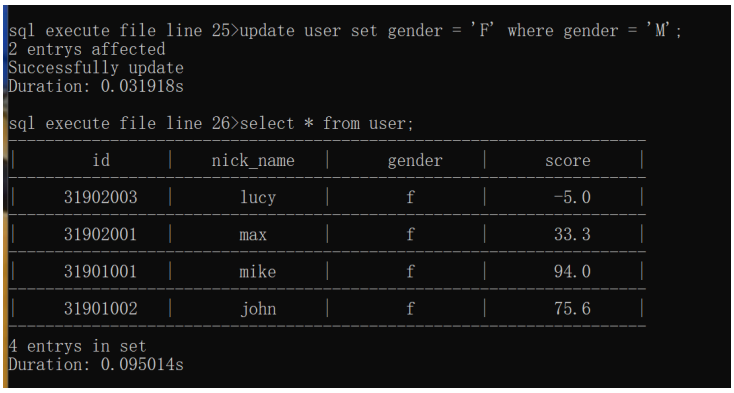


恢复数据

```
update user set nick_name = 'john' where nick_name = 'bdz';
```

单条件，多条单列更新

```
update user set gender = 'F' where gender = 'M';select * from user;
```

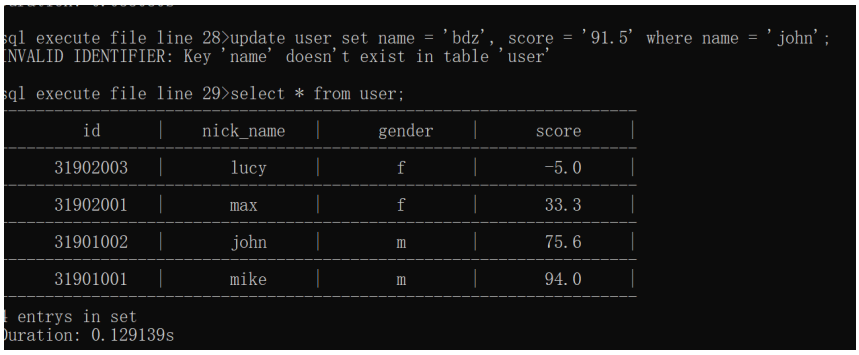


恢复数据

```
update user set gender = 'M' where score > 50;
```

单条件，单条多列更新

```
update user set name = 'bdz', score = '91.5' where name = 'john';select * from user;
```

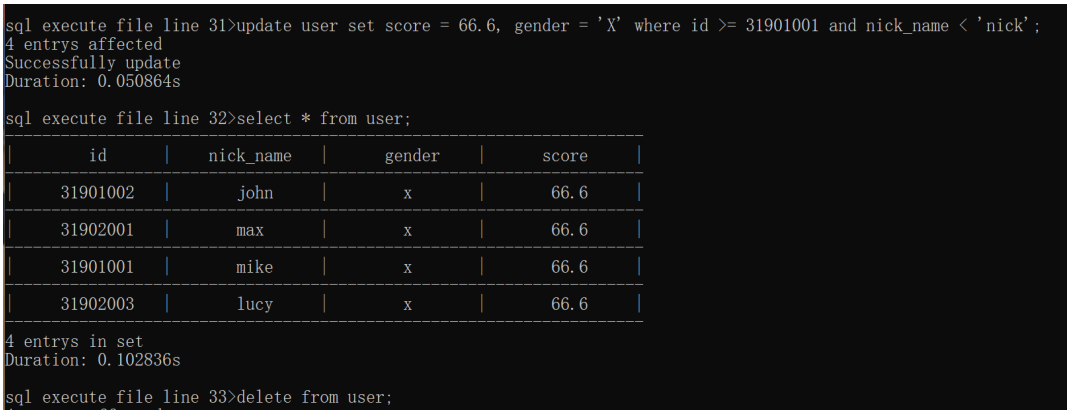


恢复数据

```
update user set name = 'john', score = '94.0' where name = 'bdz';
```

多条件，多条多列更新

```
update user set score = 66.6, gender = 'X' where id >= 31901001 and nick_name < 'nick';select * from user;
```



重建数据

```
delete from user;insert into user values (31901001, 'mike', 'M', 94.0);insert into user values (31901002, 'john', 'M', 75.6);insert into user values (31902001, 'max', 'F', 33.3);insert into user values (31902003, 'lucy', 'F', -5.0);
```

2.2 show相关语句

语句格式

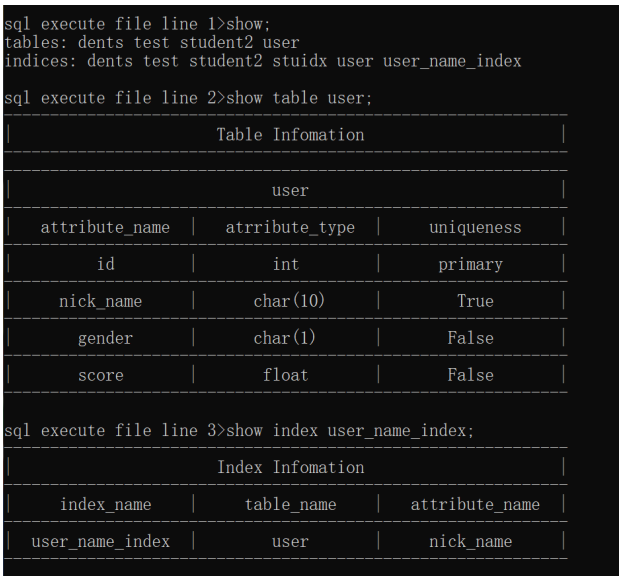
```
show;show table 表名;show index 索引名;
```

说明：

1. 展示表、索引的简略信息；
2. 展示指定表格的信息，信息包含属性名、属性值类型，是否unique/是否为主键；
3. 展示指定索引信息，信息包含索引名，表名，属性名；

测试：

```
show;
show table user;
show index user_name_index;
```



2.3 help

语句格式：

```
help;
```

说明：

展示支持的语句功能提示。

测试：

```
help;
```

```
sql execute file line 4>help;
---------------Minisql-----------------
-----Developed by Cheung, SoonWhy and N7Utb-----
Support:
- create a table
- create an index
- drop a table
- drop an index
- insert records into a table
- delete records from a table
- select from a table
- execute instructions in a file
- extra: use update instruction to update records in a table
- extra: use show instruction to view information of tables or indices
- extra: use import instruction to build a table with data from a csv file
- extra: use export instruction to output a table into a csv file
- enter "exit" or "quit" to exit Minisql
```

2.4 import

语句格式：

```
import 表名 from 文件路径 (属性名1 值属性1 可选unique指定,..., primary key(主键属性名));
```

执行导入文件的语句：

```
import import_test from ./test/test_data/test.csv (id int,name char(10)
unique,gender char(1),primary key(id));
show table import_test;
```

```
sql execute file line 5>import import_test from ./test/test_data/test.csv (id int,name char(10) unique,gender char(1),primary key(id));
Successfully create table 'import_test'
4993 rows affected
Duration: 4.088923s

sql execute file line 6>show table import_test;
-------------------------------------------------------
|                  Table Infomation                   |
-------------------------------------------------------
-------------------------------------------------------
|                    import_test                      |
-------------------------------------------------------
|  attribute_name  |  atrribute_type  |    uniqueness   |
-------------------------------------------------------
|        id        |       int        |     primary     |
-------------------------------------------------------
|       name       |     char(10)     |       True      |
-------------------------------------------------------
|      gender      |     char(1)      |      False      |
-------------------------------------------------------
```

2.5 SQL查询优化

我们在执行select和delete的时候都做了查询优化，如果条件语句中涉及已建立索引的属性，将首先利用索引做第一轮搜索，进而判断以优化查询。

比较创立索引前后的搜索时间。

```
select * from import_test where name = 'qvzmy';
create index import_test_name_index on import_test (name);
select * from import_test where name = 'qvzmy';
```

```
sql execute file line 7>select * from import_test where name = 'qvzmy';
-------------------------------------------------
|        id      |       name     |      gender    |
-------------------------------------------------
0 entrys in set
Duration: 0.007978s

sql execute file line 8>create index import_test_name_index on import_test (name);
INVALID IDENTIFIER Error: import_test_name_index index already exists

sql execute file line 9>select * from import_test where name = 'qvzmy';
-------------------------------------------------
|        id      |       name     |      gender    |
-------------------------------------------------
0 entrys in set
Duration: 0.002992s
```

2.6 output

语句格式：

```
export 路径 from 表名;
```

说明：

将指定的表格以csv的形式输出到指定路径下。

测试：

```
export ./test/test_data/user.csv from user;
```

```
sql execute file line 12>export ./test/test_data/user.csv from user;
Succesfully Output, you can find the file at: ./test/test_data/user.csv
```

Part 3 异常提示测试

在miniSQL使用过程中，有诸多异常检测提醒，包括但不限于：SYNTAX Error, INVAILD IDENTIFIER, INVALID VALUE等等，我们也对上述语句实际操作过程中可能出现的错误设置了错误排查，在这里做一个测试

3.1 SQL语法错误 SYNTAX ERROR

Type 1: 不符合SQL语法规范

```
select from user;select from userinsert into user (31901003, 'ErrorTest', 'F',
88.88);
```

> 第一句没有包含指定的列；
>
> 第二句没有标志结尾的分号
>
> 第三句没有包含values关键字

Type 2: SQL语义解析不合理

```
create table errortest (id int, name char(10), primary key (stu_id));import
import_error_test from ./test/test_data/test.csv (id int,stu_name char(10)
unique,gender char(1),primary key(id));insert into user (31901008);
```

> 第一句选择的primary key并不在表设定的属性中
>
> 第二句指定的第二个属性stu_name并不在读取的文件所含的属性中
>
> 第三句输入的值数量与对应表的属性数量不匹配

3.2 SQL参量错误 INVALID IDENTIFIER

Type 1-1: 表不存在

```
select * from fake_table;
```

> fake_table 不存在

Type 1-2: 表已经存在

```
create table user (id int, primary key (id));import user from
./test/test_data/test.csv (id int,stu_name char(10) unique,gender
char(1),primary key(id));
```

> user表已经存在

Type 2: 属性名设定不正确

```
delete from user where stu_name = 'bdz';
```

> stu_name不存在

Type 3: 索引名字设定不正确

```
create index user_name_index on user (id);
```

> 同样的索引已经创建好了

3.3 SQL参数错误 INVALID VALUE

```
insert into user values (31901008, 'blablablablablablablabla', 'M',
66.66);insert into user values (31901001, 'john', 'M', 66.66);
```

> nick_name过长
>
> unique属性重复

3.4 其他错误

除却以上正确使用过程中可能遇到的因用户不正确操作产生的警示，我们还考虑了因物理储存导致的数据储存的错误提示，大致包括：

- 表数据文件损坏或缺失
- 索引文件损坏或缺失

- 元数据文件损坏或缺失

会产生如下报错（仅展示部分），因较难通过命令行触发错误，因此这里暂不进行测试：

```
ERROR: The catalog folder is missing! The program exits and please rebuild the folder!ERROR: ALL the indices may be lost! Please check your files!
```

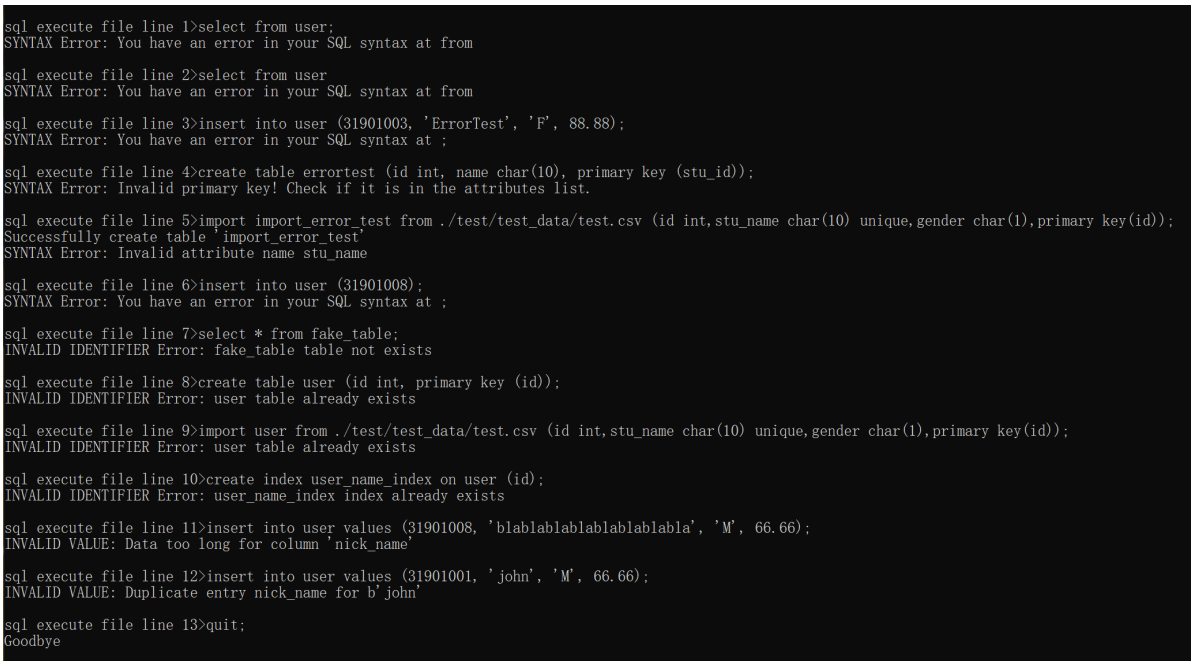
Part 4 退出

实际可用的有两种退出指令

```
quit;exit;
```

这里我们采用实验指定的quit

```
quit;
```



六、 一些备注与遇到的问题

1. 二进制文件的读写与处理：利用`struct`包
2. 模块的层次问题：为了避免循环导入，新增一个main.py的顶层，进行统一的获得标准输入及错误处理。
3. buffer与文件的一致性问题：由于本次实现的简单数据库不考虑rollback的问题，因此我们总体上采用Write-Back的策略。只有在需要确保文件为最新修改版本的时候，将buffer中的修改都提交一次。
4. 块的处理问题：buffer只负责读块与写块，怎么写与怎么读都由其他模块决定
5. 团队开发协作：利用git，保持交流