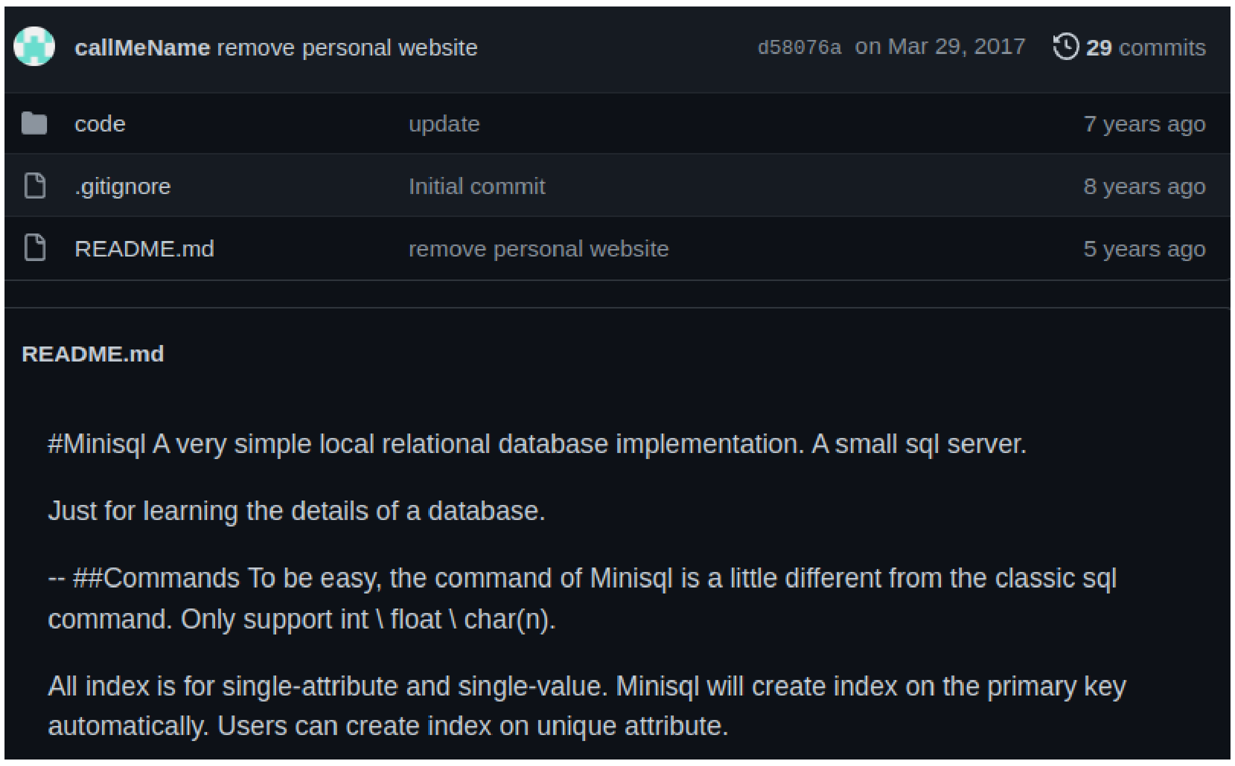
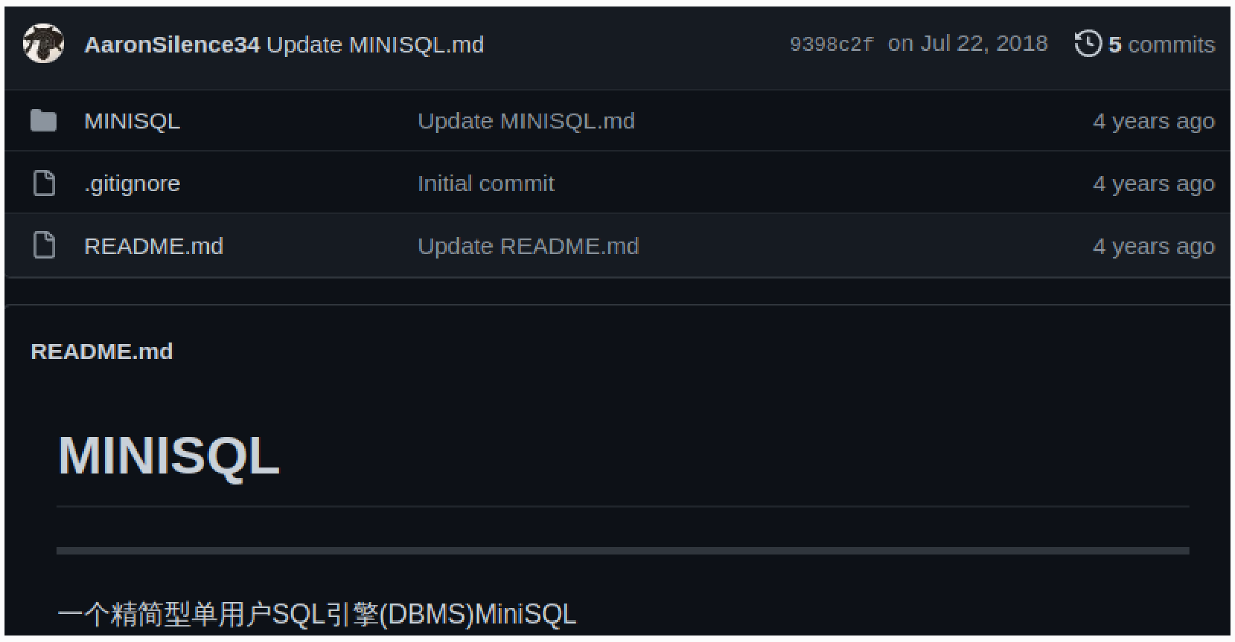
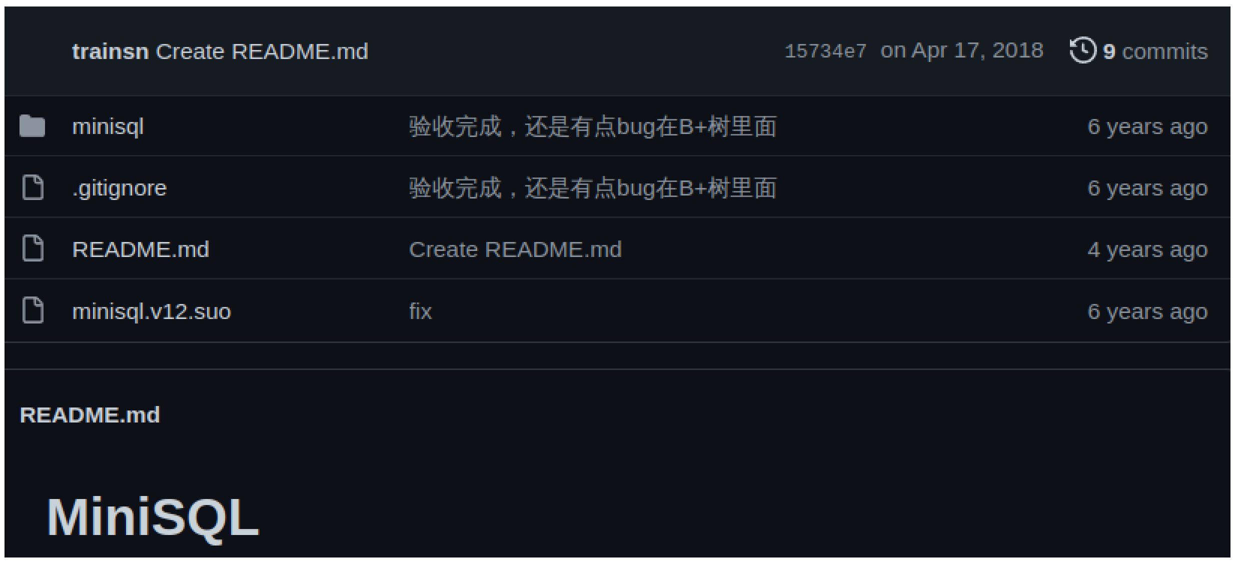
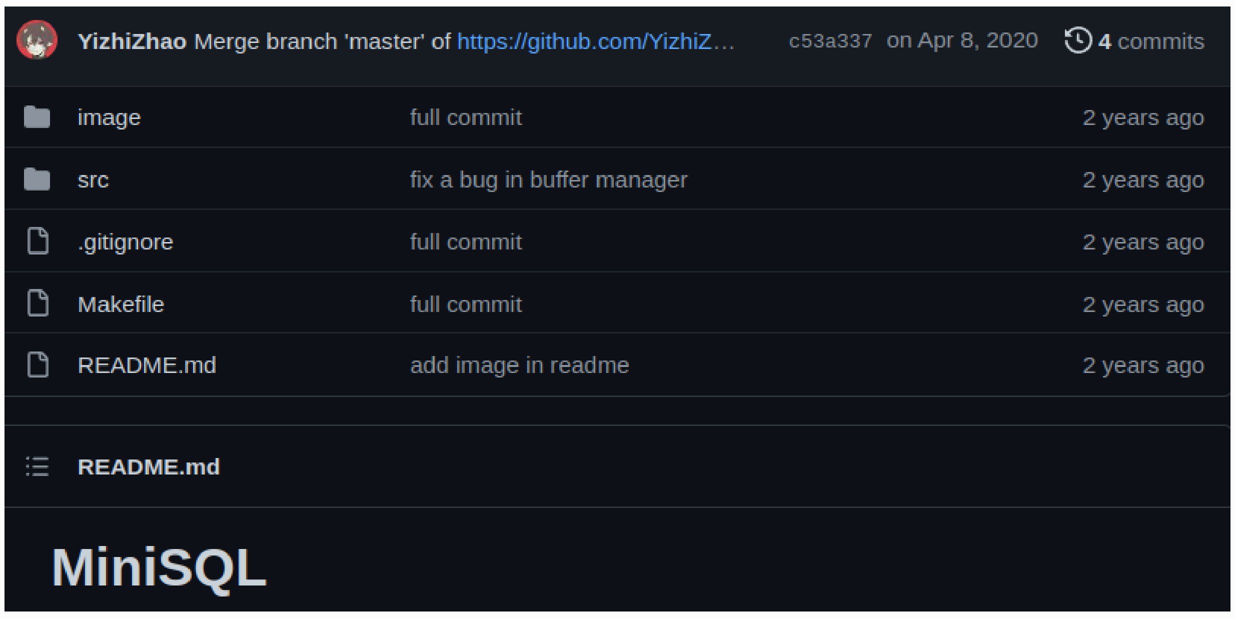
爬取了github上80个（共264个）C++ minisql项目，以默认的80%相似度为阈值检测克隆，得到结果（摘取很小一部分结果分析）：

```
38,428,1,6
38,431,1,9
40,459,1,2
40,463,1,6
40,467,1,10
40,463,38,428
60,674,40,467
60,674,1,10
```

不妨挨个分析

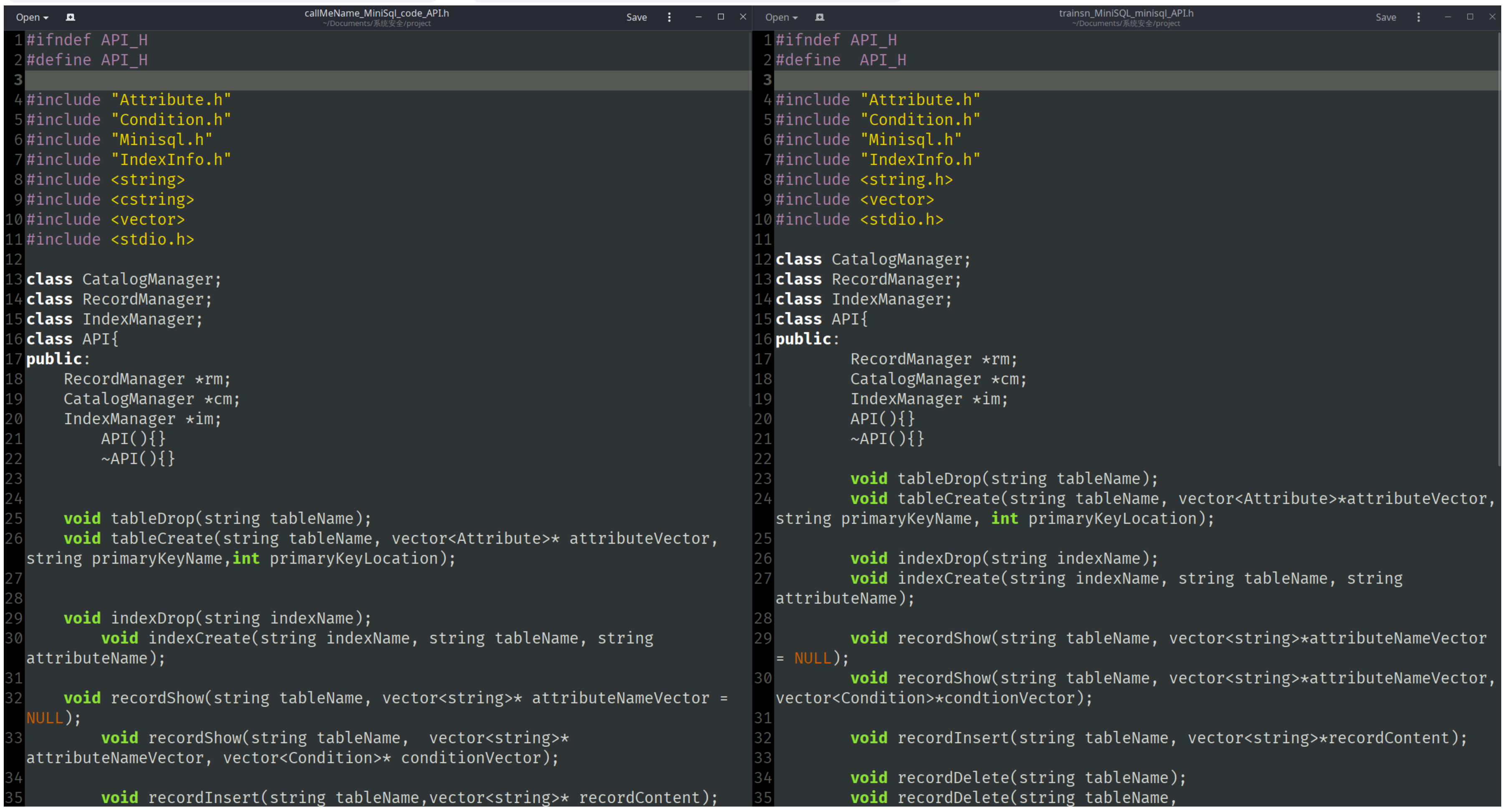
项目id和项目的对应关系在 `bookkeeping_projs/` 中

project_id	project	缩略图
1	callMeName/MiniSql	 callMeName remove personal website d58076a on Mar 29, 2017 29 commits code update 7 years ago .gitignore Initial commit 8 years ago README.md remove personal website 5 years ago README.md #Minisql A very simple local relational database implementation. A small sql server. Just for learning the details of a database. -- ##Commands To be easy, the command of Minisql is a little different from the classic sql command. Only support int \ float \ char(n). All index is for single-attribute and single-value. Minisql will create index on the primary key automatically. Users can create index on unique attribute.
38	AaronSilence34/MINISQL	 AaronSilence34 Update MINISQL.md 9398c2f on Jul 22, 2018 5 commits MINISQL Update MINISQL.md 4 years ago .gitignore Initial commit 4 years ago README.md Update README.md 4 years ago README.md MINISQL 一个精简型单用户SQL引擎(DBMS)MiniSQL
40	trainsn/MiniSQL	 trainsn Create README.md 15734e7 on Apr 17, 2018 9 commits minisql 验收完成，还是有点bug在B+树里面 6 years ago .gitignore 验收完成，还是有点bug在B+树里面 6 years ago README.md Create README.md 4 years ago minisql.v12.suo fix 6 years ago README.md MiniSQL
60	YizhiZhao/MiniSQL	 YizhiZhao Merge branch 'master' of https://github.com/YizhiZ… c53a337 on Apr 8, 2020 4 commits image full commit 2 years ago src fix a bug in buffer manager 2 years ago .gitignore full commit 2 years ago Makefile full commit 2 years ago README.md add image in readme 2 years ago README.md MiniSQL

project_id, file_id	file
1,2	code/API.h
1,6	code/BufferManager.cpp
1,9	code/CatalogManager.h
1,10	code/Condition.cpp
38,428	src/BufferManager.cpp
38,431	src/CatalogManager.h
40,459	minisql/API.h
40,463	minisql/BufferManager.cpp
40,467	minisql/Condition.cpp
60,674	lib/Condition.cpp

于是分析结果就是：

- `callMeName/MiniSql/code/API.h` 与 `trainsn/MiniSQL/minisql/API.h` 相似：

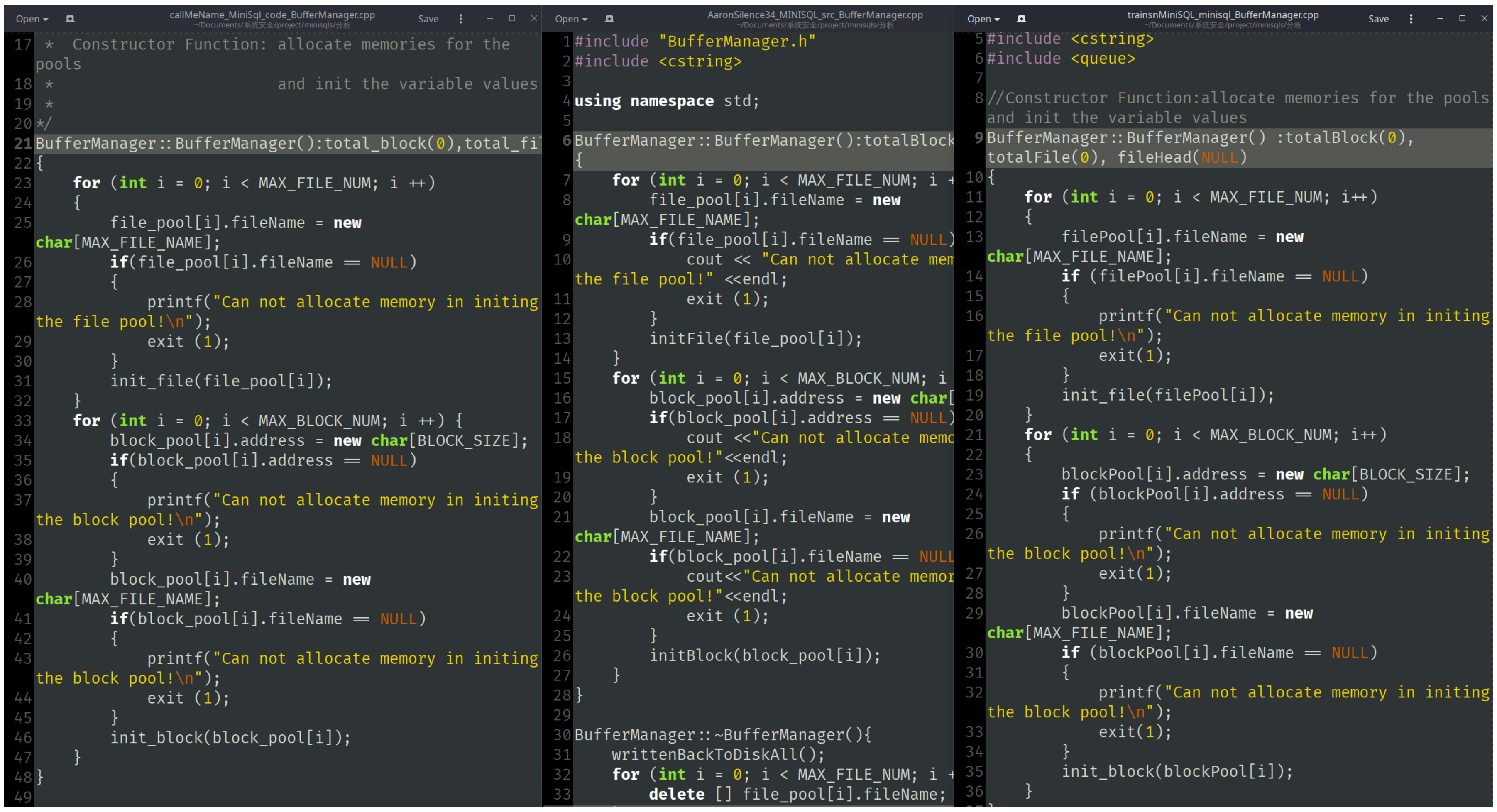

Open
callMeName_MiniSql_code_API.h
~/Documents/系统安全/project
Save
Open
trainsn_MiniSQL_minisql_API.h
~/Documents/系统安全/project
Save

```cpp
#ifndef API_H
#define API_H

#include "Attribute.h"
#include "Condition.h"
#include "Minisql.h"
#include "IndexInfo.h"
#include <string>
#include <cstring>
#include <vector>
#include <stdio.h>

class CatalogManager;
class RecordManager;
class IndexManager;
class API{
public:
    RecordManager *rm;
    CatalogManager *cm;
    IndexManager *im;
        API(){}
        ~API(){}


    void tableDrop(string tableName);
    void tableCreate(string tableName, vector<Attribute>* attributeVector, string primaryKeyName,int primaryKeyLocation);


    void indexDrop(string indexName);
        void indexCreate(string indexName, string tableName, string attributeName);

    void recordShow(string tableName, vector<string>* attributeNameVector = NULL);
        void recordShow(string tableName,  vector<string>* attributeNameVector, vector<Condition>* conditionVector);

        void recordInsert(string tableName,vector<string>* recordContent);
```

```cpp
#ifndef API_H
#define  API_H

#include "Attribute.h"
#include "Condition.h"
#include "Minisql.h"
#include "IndexInfo.h"
#include <string.h>
#include <vector>
#include <stdio.h>

class CatalogManager;
class RecordManager;
class IndexManager;
class API{
public:
        RecordManager *rm;
        CatalogManager *cm;
        IndexManager *im;
        API(){}
        ~API(){}

        void tableDrop(string tableName);
        void tableCreate(string tableName, vector<Attribute>*attributeVector, string primaryKeyName, int primaryKeyLocation);

        void indexDrop(string indexName);
        void indexCreate(string indexName, string tableName, string attributeName);

        void recordShow(string tableName, vector<string>*attributeNameVector = NULL);
        void recordShow(string tableName, vector<string>*attributeNameVector, vector<Condition>*condtionVector);

        void recordInsert(string tableName, vector<string>*recordContent);

        void recordDelete(string tableName);
        void recordDelete(string tableName,
```

《不演了》

- `callMeName/MiniSql/code/BufferManager.cpp` 与 `AaronSilence34/MINISQL/src/BufferManager.cpp` 与 `trainsn/MiniSQL/minisql/BufferManager.cpp` 相似：


Open
callMeName_MiniSql_code_BufferManager.cpp
~/Documents/系统安全/project/minisqls/分析
Save
Open
AaronSilence34_MINISQL_src_BufferManager.cpp
~/Documents/系统安全/project/minisqls/分析
Open
trainsnMiniSQL_minisql_BufferManager.cpp
~/Documents/系统安全/project/minisqls/分析
Save

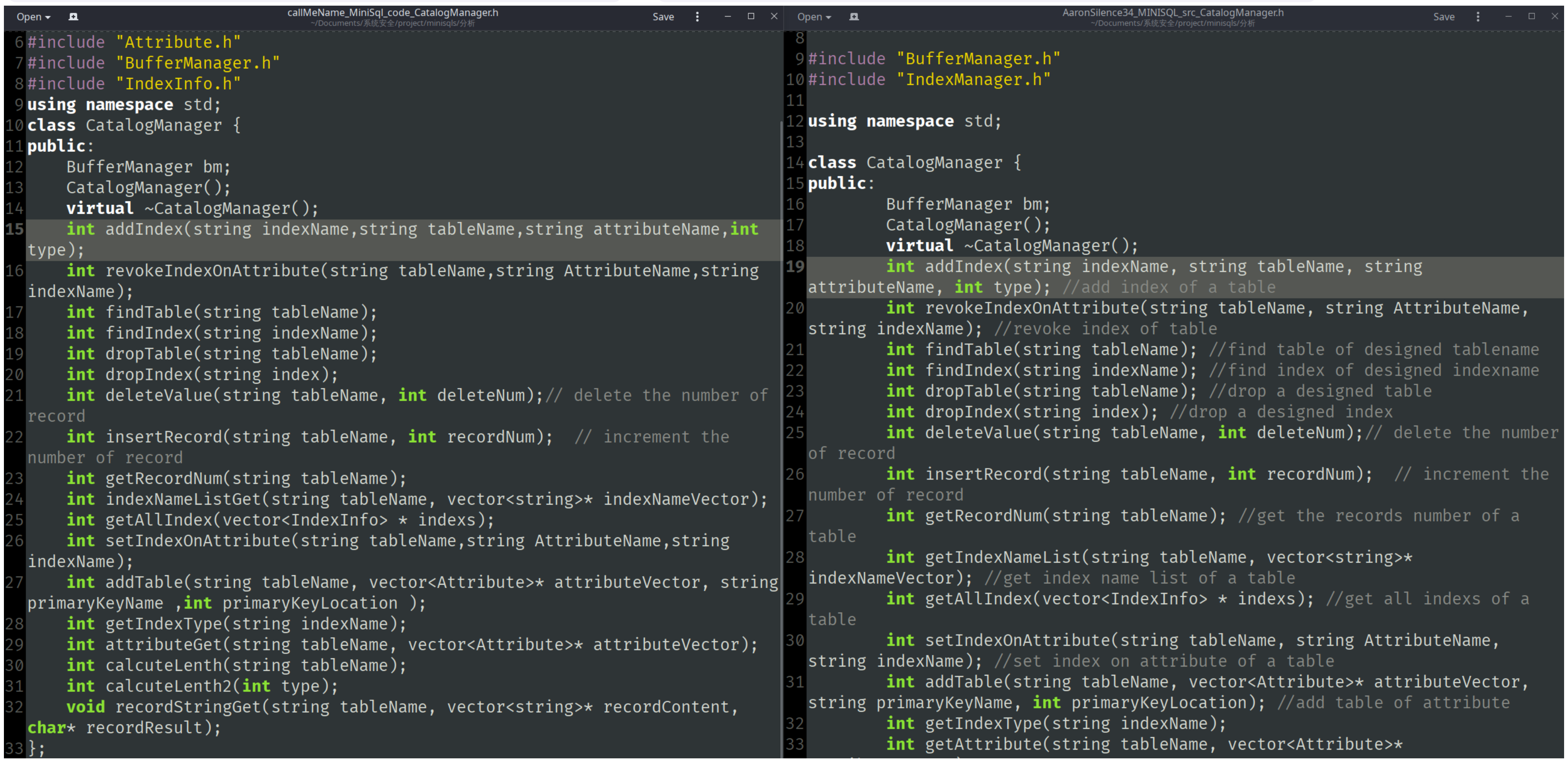
```cpp
 *  Constructor Function: allocate memories for the pools
 *                          and init the variable values
 *
 */
BufferManager::BufferManager():total_block(0),total_fi
{
    for (int i = 0; i < MAX_FILE_NUM; i ++)
    {
        file_pool[i].fileName = new char[MAX_FILE_NAME];
        if(file_pool[i].fileName == NULL)
        {
            printf("Can not allocate memory in initing the file pool!\n");
            exit (1);
        }
        init_file(file_pool[i]);
    }
    for (int i = 0; i < MAX_BLOCK_NUM; i ++) {
        block_pool[i].address = new char[BLOCK_SIZE];
        if(block_pool[i].address == NULL)
        {
            printf("Can not allocate memory in initing the block pool!\n");
            exit (1);
        }
        block_pool[i].fileName = new char[MAX_FILE_NAME];
        if(block_pool[i].fileName == NULL)
        {
            printf("Can not allocate memory in initing the block pool!\n");
            exit (1);
        }
        init_block(block_pool[i]);
    }
}
```

```cpp
#include "BufferManager.h"
#include <cstring>

using namespace std;

BufferManager::BufferManager():totalBlock
{
    for (int i = 0; i < MAX_FILE_NUM; i +
        file_pool[i].fileName = new char[MAX_FILE_NAME];
        if(file_pool[i].fileName == NULL)
            cout << "Can not allocate mem the file pool!" <<endl;
            exit (1);
        }
        initFile(file_pool[i]);
    }
    for (int i = 0; i < MAX_BLOCK_NUM; i
        block_pool[i].address = new char[
        if(block_pool[i].address == NULL)
            cout <<"Can not allocate memo the block pool!"<<endl;
            exit (1);
        }
        block_pool[i].fileName = new char[MAX_FILE_NAME];
        if(block_pool[i].fileName == NULL
            cout<<"Can not allocate memor the block pool!"<<endl;
            exit (1);
        }
        initBlock(block_pool[i]);
    }
}

BufferManager::~BufferManager(){
    writtenBackToDiskAll();
    for (int i = 0; i < MAX_FILE_NUM; i +
        delete [] file_pool[i].fileName;
```

```cpp
#include <cstring>
#include <queue>

//Constructor Function:allocate memories for the pools and init the variable values
BufferManager::BufferManager() :totalBlock(0), totalFile(0), fileHead(NULL)
{
    for (int i = 0; i < MAX_FILE_NUM; i++)
    {
        filePool[i].fileName = new char[MAX_FILE_NAME];
        if (filePool[i].fileName == NULL)
        {
            printf("Can not allocate memory in initing the file pool!\n");
            exit(1);
        }
        init_file(filePool[i]);
    }
    for (int i = 0; i < MAX_BLOCK_NUM; i++)
    {
        blockPool[i].address = new char[BLOCK_SIZE];
        if (blockPool[i].address == NULL)
        {
            printf("Can not allocate memory in initing the block pool!\n");
            exit(1);
        }
        blockPool[i].fileName = new char[MAX_FILE_NAME];
        if (blockPool[i].fileName == NULL)
        {
            printf("Can not allocate memory in initing the block pool!\n");
            exit(1);
        }
        init_block(blockPool[i]);
    }
```

《一眼丁真》

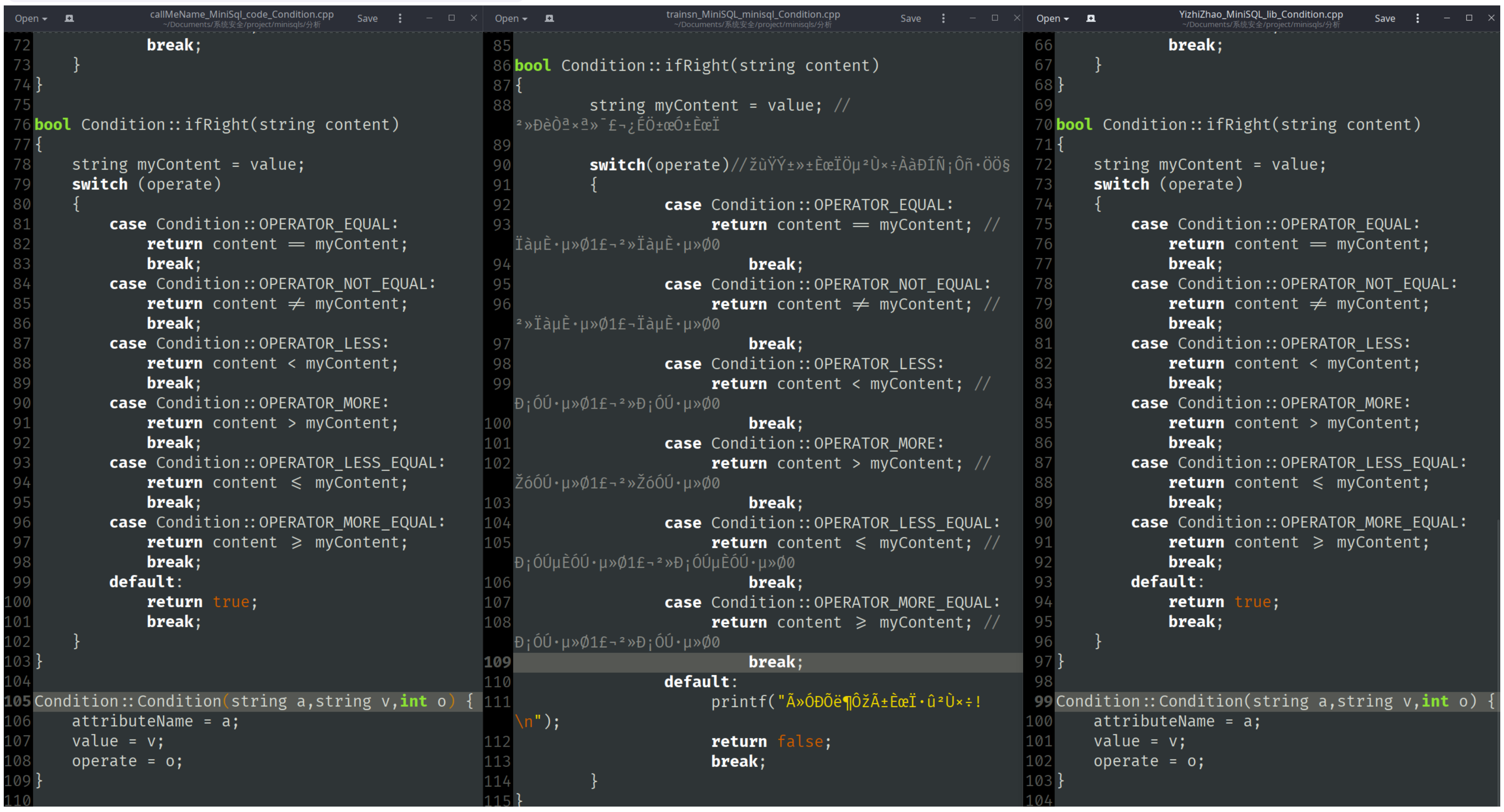
- `callMeName/MiniSql/code/CatalogManager.h` 与 `AaronSilence34/MINISQL/src/CatalogManager.h` 相似：


Open
callMeName_MiniSql_code_CatalogManager.h
~/Documents/系统安全/project/minisqls/分析
Save
Open
AaronSilence34_MINISQL_src_CatalogManager.h
~/Documents/系统安全/project/minisqls/分析
Save

```cpp
#include "Attribute.h"
#include "BufferManager.h"
#include "IndexInfo.h"
using namespace std;
class CatalogManager {
public:
    BufferManager bm;
    CatalogManager();
    virtual ~CatalogManager();
    int addIndex(string indexName,string tableName,string attributeName,int type);
    int revokeIndexOnAttribute(string tableName,string AttributeName,string indexName);
    int findTable(string tableName);
    int findIndex(string indexName);
    int dropTable(string tableName);
    int dropIndex(string index);
    int deleteValue(string tableName, int deleteNum);// delete the number of record
    int insertRecord(string tableName, int recordNum);  // increment the number of record
    int getRecordNum(string tableName);
    int indexNameListGet(string tableName, vector<string>* indexNameVector);
    int getAllIndex(vector<IndexInfo> * indexs);
    int setIndexOnAttribute(string tableName,string AttributeName,string indexName);
    int addTable(string tableName, vector<Attribute>* attributeVector, string primaryKeyName ,int primaryKeyLocation );
    int getIndexType(string indexName);
    int attributeGet(string tableName, vector<Attribute>* attributeVector);
    int calcuteLenth(string tableName);
    int calcuteLenth2(int type);
    void recordStringGet(string tableName, vector<string>* recordContent, char* recordResult);
};
```

```cpp

#include "BufferManager.h"
#include "IndexManager.h"

using namespace std;

class CatalogManager {
public:
        BufferManager bm;
        CatalogManager();
        virtual ~CatalogManager();
        int addIndex(string indexName, string tableName, string attributeName, int type); //add index of a table
        int revokeIndexOnAttribute(string tableName, string AttributeName, string indexName); //revoke index of table
        int findTable(string tableName); //find table of designed tablename
        int findIndex(string indexName); //find index of designed indexname
        int dropTable(string tableName); //drop a designed table
        int dropIndex(string index); //drop a designed index
        int deleteValue(string tableName, int deleteNum);// delete the number of record
        int insertRecord(string tableName, int recordNum);  // increment the number of record
        int getRecordNum(string tableName); //get the records number of a table
        int getIndexNameList(string tableName, vector<string>* indexNameVector); //get index name list of a table
        int getAllIndex(vector<IndexInfo> * indexs); //get all indexs of a table
        int setIndexOnAttribute(string tableName, string AttributeName, string indexName); //set index on attribute of a table
        int addTable(string tableName, vector<Attribute>* attributeVector, string primaryKeyName, int primaryKeyLocation); //add table of attribute
        int getIndexType(string indexName);
        int getAttribute(string tableName, vector<Attribute>*
```

- `callMeName/MiniSql/code/Condition.cpp` 与 `trainsn/MiniSQL/minisql/Condition.cpp` 与 `YizhiZhao/MiniSQL/lib/Condition.cpp` 相似：

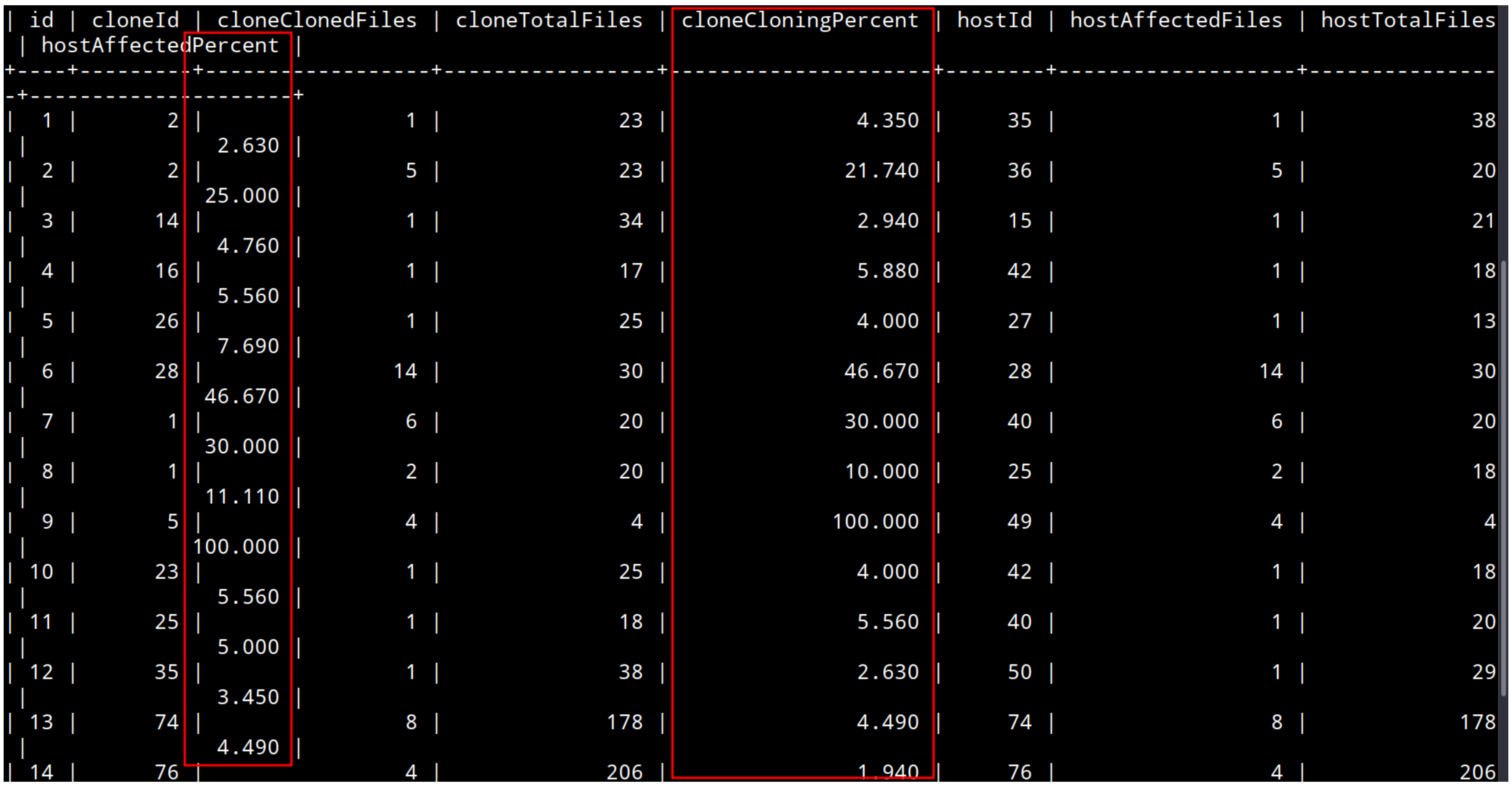

Open
callMeName_MiniSql_code_Condition.cpp
~/Documents/系统安全/project/minisqls/分析
Save
Open
trainsn_MiniSQL_minisql_Condition.cpp
~/Documents/系统安全/project/minisqls/分析
Save
Open
YizhiZhao_MiniSQL_lib_Condition.cpp
~/Documents/系统安全/project/minisqls/分析
Save

```cpp
                break;
        }
}

bool Condition::ifRight(string content)
{
    string myContent = value;
    switch (operate)
    {
        case Condition::OPERATOR_EQUAL:
            return content == myContent;
            break;
        case Condition::OPERATOR_NOT_EQUAL:
            return content != myContent;
            break;
        case Condition::OPERATOR_LESS:
            return content < myContent;
            break;
        case Condition::OPERATOR_MORE:
            return content > myContent;
            break;
        case Condition::OPERATOR_LESS_EQUAL:
            return content <= myContent;
            break;
        case Condition::OPERATOR_MORE_EQUAL:
            return content >= myContent;
            break;
        default:
            return true;
            break;
    }
}

Condition::Condition(string a,string v,int o)
{
    attributeName = a;
    value = v;
    operate = o;
}
```

```cpp

bool Condition::ifRight(string content)
{
        string myContent = value; // ²»ÐèÒª×ª»¯£¬¿ÉÖ±œÓ±ÈœÏ

        switch(operate)//žùŸÝ±»±ÈœÏÖµ²Ù×÷ÀàÐÍ¡Ôñ·ÖÖ§
        {
                case Condition::OPERATOR_EQUAL:
                        return content == myContent; // ÏàµÈ·µ»Ø1£¬²»ÏàµÈ·µ»Ø0
                        break;
                case Condition::OPERATOR_NOT_EQUAL:
                        return content != myContent; // ²»ÏàµÈ·µ»Ø1£¬ÏàµÈ·µ»Ø0
                        break;
                case Condition::OPERATOR_LESS:
                        return content < myContent; // Ð¡ÓÚ·µ»Ø1£¬²»Ð¡ÓÚ·µ»Ø0
                        break;
                case Condition::OPERATOR_MORE:
                        return content > myContent; // ŽóÓÚ·µ»Ø1£¬²»ŽóÓÚ·µ»Ø0
                        break;
                case Condition::OPERATOR_LESS_EQUAL:
                        return content <= myContent; // Ð¡ÓÚµÈÓÚ·µ»Ø1£¬²»Ð¡ÓÚµÈÓÚ·µ»Ø0
                        break;
                case Condition::OPERATOR_MORE_EQUAL:
                        return content >= myContent; // Ð¡ÓÚ·µ»Ø1£¬²»Ð¡ÓÚ·µ»Ø0
                        break;
                default:
                        printf("Ã»ÓÐÕë¶ÔžÃ±ÈœÏ·û²Ù×÷!\n");
                        return false;
                        break;
        }
}
```

```cpp
                break;
        }
}

bool Condition::ifRight(string content)
{
    string myContent = value;
    switch (operate)
    {
        case Condition::OPERATOR_EQUAL:
            return content == myContent;
            break;
        case Condition::OPERATOR_NOT_EQUAL:
            return content != myContent;
            break;
        case Condition::OPERATOR_LESS:
            return content < myContent;
            break;
        case Condition::OPERATOR_MORE:
            return content > myContent;
            break;
        case Condition::OPERATOR_LESS_EQUAL:
            return content <= myContent;
            break;
        case Condition::OPERATOR_MORE_EQUAL:
            return content >= myContent;
            break;
        default:
            return true;
            break;
    }
}

Condition::Condition(string a,string v,int o) {
    attributeName = a;
    value = v;
    operate = o;
}
```

导入数据库之后的克隆结果:



```
| id | cloneId | cloneClonedFiles | cloneTotalFiles | cloneCloningPercent | hostId | hostAffectedFiles | hostTotalFiles
 | hostAffectedPercent |
+----+---------+------------------+-----------------+---------------------+--------+-------------------+---------------
-+---------------------+
|  1 |       2 |                1 |              23 |               4.350 |     35 |                 1 |             38
 |               2.630 |
|  2 |       2 |                5 |              23 |              21.740 |     36 |                 5 |             20
 |              25.000 |
|  3 |      14 |                1 |              34 |               2.940 |     15 |                 1 |             21
 |               4.760 |
|  4 |      16 |                1 |              17 |               5.880 |     42 |                 1 |             18
 |               5.560 |
|  5 |      26 |                1 |              25 |               4.000 |     27 |                 1 |             13
 |               7.690 |
|  6 |      28 |               14 |              30 |              46.670 |     28 |                14 |             30
 |              46.670 |
|  7 |       1 |                6 |              20 |              30.000 |     40 |                 6 |             20
 |              30.000 |
|  8 |       1 |                2 |              20 |              10.000 |     25 |                 2 |             18
 |              11.110 |
|  9 |       5 |                4 |               4 |             100.000 |     49 |                 4 |              4
 |             100.000 |
| 10 |      23 |                1 |              25 |               4.000 |     42 |                 1 |             18
 |               5.560 |
| 11 |      25 |                1 |              18 |               5.560 |     40 |                 1 |             20
 |               5.000 |
| 12 |      35 |                1 |              38 |               2.630 |     50 |                 1 |             29
 |               3.450 |
| 13 |      74 |                8 |             178 |               4.490 |     74 |                 8 |            178
 |               4.490 |
| 14 |      76 |                4 |             206 |               1.940 |     76 |                 4 |            206
```