Proj1：

实验名：**聚集存储实现**

实验目的：

根据磁盘管理的知识设计并实现一个磁盘存储系统

* + 插入记录（老师代码已实现）-insertRecord(std::vector<struct iovec> &iov);
  + 修改记录（自己实现）-updateRecord(std::vector<struct iovec> &iov);
  + 删除记录 (自己实现)
  + 枚举记录（即查询记录，老师代码已实现）- searchRecord(void \*key, size\_t size);

插入：

std::pair<bool, unsigned short>

DataBlock::insertRecord(std::vector<struct iovec> &iov)

{

RelationInfo \*info = table\_->info\_;

unsigned int key = info->key;

DataType \*type = info->fields[key].type;

// 先确定插入位置

unsigned short index =

type->search(buffer\_, key, iov[key].iov\_base, iov[key].iov\_len);

// 比较key

Record record;

if (index < getSlots()) {

Slot \*slots = getSlotsPointer();

record.attach(

buffer\_ + be16toh(slots[index].offset),

be16toh(slots[index].length));

unsigned char \*pkey;

unsigned int len;

record.refByIndex(&pkey, &len, key);

if (memcmp(pkey, iov[key].iov\_base, len) == 0) // key相等不能插入

return std::pair<bool, unsigned short>(false, -1);

}

// 如果block空间足够，插入

size\_t blen = getFreeSize(); // 该block的富余空间

unsigned short actlen = (unsigned short) Record::size(iov);

unsigned short alignlen = ALIGN\_TO\_SIZE(actlen);

unsigned short trailerlen =

ALIGN\_TO\_SIZE((getSlots() + 1) \* sizeof(Slot) + sizeof(unsigned int)) -

ALIGN\_TO\_SIZE(getSlots() \* sizeof(Slot) + sizeof(unsigned int));

if (blen < actlen + trailerlen)

return std::pair<bool, unsigned short>(false, index);

// 分配空间

std::pair<unsigned char \*, bool> alloc\_ret = allocate(actlen, index);

// 填写记录

record.attach(alloc\_ret.first, actlen);

unsigned char header = 0;

record.set(iov, &header);

// 重新排序

if (alloc\_ret.second) reorder(type, key);

return std::pair<bool, unsigned short>(true, index);

}