数据库开发文档

519030910322 李至丹

实现功能

实现train、ticket和user信息存储功能。实现上使用B+树实现。文件存储。

B+树接口(bptree\_fixed.hpp)

1.基本接口。

(1) 构造函数：sjtu::bptree<Key, value\_t, K, Compare> t(file1, file2)

其中，Key表示关键字的类型，value\_t是对应的元素值的类型，K表示B+树默认的节点空间，默认为4096，Compare为Key的比较方式，默认为std::less<Key>。

数据存储在file1当中，索引为file2。

(2) 插入函数：void insert(Key k, value\_t v)；

在B+树中插入一个关键字为k，值为v的元素。如果key已经存在，则将对应的值修改该为v。

(3) 修改函数：bool modify(Key k, value\_t v)；

在B+树中将关键字为k的元素值修改为v；如果不存在，则插入进去。返回值为是否修改成功。

(4) 删除函数：void erase(Key k)；

删除关键字为k的元素。必须保证关键字为k的元素存在，否则，抛出异常：erase leaf error!和erase node error!。

(5) 计数函数：int count(Key k)；

判断关键字为k的元素是否存在。存在返回1，否则返回0。

(6) 取值函数：(const) value\_t at(Key k)；

将关键字为k的元素值取出。如果不存在，则返回value\_t的空元素。

(7) 判空函数：bool empty()；

判断B+树是否为空树。如果是，则返回1；否则返回0。

(8) 编译函数：void traverse()；

输出B+树当中的所有节点信息。

2.迭代器接口。

(1) 基本迭代器类型：iterator和const\_iterator；表示B+树存储的一个信息。

(a) 使用运算符’\*’和函数get\_key()得到存储位置的关键字。

(b) 使用运算符’++’和’--’来遍历。

(c) 通过运算符’==’和‘!=’来判断是否相等。

(2) 查询函数：iterator lower\_bound(Key k)和const\_iterator clower\_bound(Key k)；

找到第一个关键字不小于k的元素。

(3) 边界函数：iterator begin(), iterator end()和const\_iterator cbegin(), const\_iterator cend()；

分别表示第一个节点和最后一个节点。默认B+树的范围为左闭右开。

vector接口(file\_vector.hpp)

1.基本接口。

(1) 构造函数：SJTU::vector<T, filename> v；

T为vector当中的元素类型，filename为存储元素的文件。

(2) 判空函数和大小：bool empty()和size\_t size()；

判断vector是否为空的函数和返回vector当中的元素个数。

(3) 清空：void clean()；

将vector当中的元素清空，同时删除对应的存储文件。

(4) 初始化函数：void init()；

从filename当中读取元素。

(5) 修改函数：void modify(size\_t index, T value)；

将第index个值修改为value。如果不在范围之内，抛出异常。

(6) 插入函数：void push\_back(T value)；

将value插入到vector的末端。

(7) 删除函数：void pop\_back()和void erase(size\_t index)；

将尾部删除和将第index个元素删除。

(8) 读取函数：T operator [](size\_t index)。

得到第index个元素的值。

(9) 头尾函数：T front()和T back()；

得到vector的第0个元素和最后一个元素。

内存缓存(memory\_pool.hpp)

基本实现：LRU cache算法，hash+链表实现。

1.基本接口

(1) 读取函数：void buffer\_read(void\*value, size\_t pos, size\_t size, char \*filename)；

从文件名为filename的文件的第pos个位置，读取长度为size字节的内容，存储在value当中。

(2) 写入函数：void buffer\_write(void\*value, size\_t pos, size\_t size, char \*filename)；

将value中的size字节的内容，写入文件名为filename的文件之中，起始位置为pos。