SimpleDB　Lab1（数据访问与存储）

实验环境

MacBook Air

Cpu:1.6GHz Intel core i5

Mem:8GB 1600MHz DDR3

HardDisk:SSD

eclipse Version: Mars.2

JavaSE-1.6

一、关键类的实现

1、Catalog

该类用于为数据库存放的每个 table 建立一个目录，成员变量有：

- int [] tableID; 保存表的 ID；
- String [] tableName; 保存表名；
- TupleDesc [] tableTD; 保存表中列的描述类；
- DbFile [] tableFile; 保存表对应的文件；
- int [] isTable; 判断某张表是否在数据库中有效，0 为无效，1 为有效。

一个 catelog 实例的形式如下表：

tableID	0	1	2	3	4	…
tableName	Student	Class	Teacher	Book	classroom	…
tableTD	td0	td1	td2	td3	td4	…
tableFile	File0	File1	File2	File3	File4	…
isTable	1	1	1	0	1	…

主要成员方法：

- public Catalog()，构造函数，该函数中将表的有效位全部置 0；
- public void addTable(DbFile file, String name)，增加一张新表；
- public int getTableId(String name)，通过表名找到表 ID；
- public TupleDesc getTupleDesc(int tableid)，通过表 ID 找到列的描述类；
- public void clear()，清空所有表，即表的有效位全部置 0；

2、TupleDesc

该类用于表中对列进行描述，一个 TupleDesc 可能是实例形式如下：

fieldID	0	1	2
fieldName	null	null	null
fieldSize	4	4	4
fieldType	INT_TYPE	INT_TYPE	INT_TYPE
num	3	3	3

比如这个 TupleDesc 可以描述(22, 32, 15)这样的一个元组，num 用于说明列的个数。

3、Tuple

Tuple 类用于描述一条记录，变量有：

- private RecordId recordId，对应的 RecordId；
- private int num，数据域的个数；
- private Field [] field，数据域
- private int [] attr，记录对应的数据域是 int 型还是 string 型。

主要成员方法：

- public TupleDesc getTupleDesc()，返回该记录的 TupleDesc；
- public RecordId getRecordId()以及 public void setRecordId()，分别对 TupleRed 返回和赋值
- public void setField(int i, Field f)，对 field[i]进行赋值；
- public Field getFiled(int i)，返回 field[i]；

4、HeapPageId

HeapPageId 类用于标识 HeapPage，其成员变量包括：

- private int tableID，对应 HeapPage 所在表的 ID；
- private int pid，对应 HeapPage 所在表中的第几页；

主要成员方法：

- public int getTableId()，返回 tableID；
- public int pageno()，返 pid；
- public int hashCode()，返回哈希值，如：tableID*1000 + pid；
- public boolean equals(Object o)，判断相等关系，相等返回 true，不相等返回 false。

5、HeapPage

HeapPage 是 Page 接口的实现类，是使用了堆方式存放数据的页，HeapPage 采用了一种简单的数据存放方式：哪里有空就把记录存在哪里。它的成员变量包括：

- private HeapPageId pid; //用 HeapPageId 标识该页；
- private TupleDesc td; //HeapPage 所在表对应的 TupleDesc；
- private byte header[]; //header 数组用 bitmap 方式标识了 slot 是否可用；
- private Tuple tuples[]; //页中的记录；
- private int numSlots; //页中的 slot 的数量；
- private int emptySlots; //页中空 slot 的数量；
- private boolean dirty; //脏位；
- private TransactionId dirtyTid; //最近将 dirty 置 1 的事务 ID

主要成员方法：

- private int getNumTuples()，计算页中 slot 的数目。
- private int getHeaderSize()，计算 header 数组的长度。

- public getId()，返回 pid
- public getSlot(int i)，tuples[i]。
- public Iterator<Tuple> iterator()，迭代器，用于遍历 HeapPage 中的 Tuple。
- public void printTuplesInPage()，为了方便调试，用该函数打印出该页中所有的记录。

6、HeapFile

HeapFile 是堆文件类，用堆存储的方法记录文件中的页。

成员变量：

- private File f;　　//关联文件对象；
- private TupleDesc td;　　//对应表的 TupleDesc；
- private int pageNum;　　//存放的 HeapPage 的数量；
- private HeapPageId [] hpID;　　//每个 HeapPage 的 HeapPageId；

主要成员方法：

- public getFile()，返回成员变量 f；
- public getTupleDesc()，返回成员变量 td；
- public getId()，类似 RecordId 的 hashCode，这里采用 javadoc 推荐的 f.getAbsoluteFile().getHashCode()；
- public readPage(PageId pid)，实现页的读取，细节请参见困难及解决方案。
- public int numPages()，返回数据中存放页的数目。计算公式为：((int)storFile.length() + BufferPool.PAGE_SIZE – 1)/ BufferPool.PAGE_SIZE。

7、seqScan 类在 Lab2 中详细介绍。

二、遇到的困难及解决方法

1、理解 SimpleDB 中数据组织形式和类的对应关系是 Lab1 的一个难点，其中几个关键点为：

（1）在 simpleDB 中，数据库中的一个 table 完全存放在一个 file(HeapFile)中，而没有将 table 拆分成几份分别存放在多个 file 中；

（2）一个 file(HeapFile)中可以包含多个 page(HeapPage)，file 中是否含有某个 page 是通过 HeapPageId 数组标识的，例如看某页是否在文件中，需要查找 HeapPageId[]中是否含有该页的 ID；

（3）HeapPage 中存储了一系列记录值，每个记录存储在页中的一个 slot 里，header[]数组通过 bitmap 的方式标识对应的 slot 是否有效。某个 slot 对应的 bit 位如果为 0，则说明该 slot 为空，或者曾经有值，但已被删除，如果 bit 位为 1，说明该 slot 中存有记录并且有效。

2、HeapPage 中 numTuples 的计算。

numTuples = (int)Math.floor((BufferPool.PAGE_SIZE*8) / (tupleSize * 8 + 1));

tupleSize * 8 计算出每个记录的大小，单位是 bit；
tupleSize * 8 + 1 计算出每个记录连同记录对应 bitmap 中一位的大小，单位是 bit；
BufferPool.PAGE_SIZE*8 计算出 BufferPool 中一页的大小，单位是 bit；
对((BufferPool.PAGE_SIZE*8) / (tupleSize * 8 + 1)结果下取整计算出一页中能存放记录的最大数量；

3、HeapPage 中 header[]数组的控制。
header[]数组中的每一个 bit 对应一个记录，要注意的是在某一个 header[i]中，bit 位是逆序存放的，即低位存放靠前的记录，高位存放靠后的记录，例如在空页中新增 11 条记录后，header[0] = 1111 1111b, header[1] = 0000 0111b。

4、HeapPage 中的 getHeaderSize()函数。
计算方法为：numSlots/8 后上取整

5、HeapFile 中的 readPage()函数。
通过 f 生成 RandomAccessFile 对象，先跳过 pid.pageno()*BufferPool.PAGE_SIZE 个 byte，再读取 BufferPool.PAGE_SIZE 个 byte 存放到一个 byte 数组中，通过 HeapPage 的构造函数构造一个新的对象，而后返回。