

Write Up for acmdb-lab1

在 lab1 中，我们实现了数据库中一些基本的数据储存类，以及基本的迭代运算。以下将按照 Exercise 的顺序，对实验进行讨论。

Exercise 1 (TupleDesc.java, Tuple.java)

TupleDesc 是由一系列 field 的描述信息构成的，因此考虑在 TupleDesc 类中额外定义一个 TDIItem 类，一个 TupleDesc 的属性 tdlItems 则由多个 TDIItem 的对象构成。TDIItem 类包含 fieldName 和 fieldType 两个属性，因此 TupleDesc 的构造函数在给定一个类型的 Array 和一个名称的 Array 时，便可以通过构造 TDIItem 的对象来实现。基于这样的定义，之后的函数实现就相对比较简单了。

在比较对象相等的函数中，首先需要看给定对象是否属于同一个类，只有属于同一个类方可进行进一步的比较。然后是先比较两者的大小，再比较两者的具体属性内容是否相等。之后的类实现中也有很多需要比较对象相等的函数，思路类似，不再赘述。

Exercise 2 (Catalog.java)

Catalog 类追踪在 database 中所有的 table。根据 addTable 函数所给出的参数列表，首先定义出 Table 的属性。根据下面的类方法，我们发现 Catalog 需要实现利用 table 的名称映射到其 id，也需要利用 id 索引到某个 table（从而得到其 file 属性，以及 file 属性对应的 TupleDesc），因此定义两个 Map 类型的属性，分别为 id_to_table, tname_to_id。

Exercise 3 (BufferPool.java)

BufferPool 类管理内存页的读写，额外添加了 num_pages 和 pid_to_page。在 getPage 方法中，如果 BufferPool 已满则报错，若不包含该 page 则添加进 bufferpool 中，最后根据索引读取相应的 page。

Exercise 4 (HeapPageId.java, RecordId.java, HeapPage.java)

HeapPageId 和 RecordId 的实现相对较为基础，不做赘述。

HeapPage 类中的 getNumTuples, getHeaderSize 方法的计算方式为 instruction 文档中所给出的公式。isSlotUsed 方法中，先计算 byte 的位置再计算 bit 的位置，然后通过移位运算符判断该 bit 是否为 1，该方法有着较广的应用范围，在给定的框架和 getNumEmptySlots, iterator 方法中均有使用。

Exercise 5 (HeapFile.java)

根据提示，使用 random_access_file 索引到该 page 所对应的 bytes 进行读取。观察到 iterator 方法的返回值类型是 DbFileIterator 而非 iterator，因此在该类中根据 DbFileIterator 所给出的模板构建了 HeapFileIterator 类，并逐个实现模板中给出的方法。实现了这个类之后，

在 `iterator` 方法中直接构建这个类的一个实例即可。

Exercise 6 (SeqScan.java)

这个类的实现也较为简单，相对较为复杂的 `getTupleDesc` 方法中通过循环加上 `.` 符号即可。

总结

虽然 lab1 总体实现的功能并不算复杂，但我也依然花了不少的时间在这个 lab 上，特别是 Exercise 5 和刚刚起步上手的时候，因为以前完全没有接触过 java 这个语言，虽然有 OOP 的编程思想但是毕竟很多功能都不了解，还是花了相当多的时间用来熟悉。