

# 实验报告

## 一、实验环境

Ubuntu 16.04

Jdk 1.8

Ant 1.9.6

## 二、实验流程

lab1 主要是实现了整个数据库的整体架构，做完后对数据库的组织有了更多的认识。Lab1 主要涉及的部分：Tuple（元组）、TupleDesc（元组描述信息）、Catalog（目录）、BufferPool（缓冲池）、HeapPage（数据页）、HeapFile（disk 上的文件）、SeqScan（全表顺序扫描）。

### Exercise1: Fields and Tuples

Tuple 是表中的一条记录，包含 td, rid, fields 三个属性，其它的就是一些构造器，get 方法和迭代器等。Field 是各个字段的具体内容。TupleDesc 是各个字段对应的 schema 描述信息，包括字段类型和字段名，在代码中通过 TDItem 来表示多个字段的描述信息。需要注意的是在实现 equals 等函数要注意判断输入值是否合理。

### Exercise2: Catalog

Catalog 用来存储各个表的 id 与其 name 的映射。我在这里定义了一个 Table 类，来存放一张表格的信息，然后一个 Catalog 含有多张表格。

### Exercise3: BufferPool

BufferPool 负责缓存内存中最近从磁盘读取的页面。这样下次查询相同的数据时可以直接从内存中读取数据进行处理，比从磁盘中读取快。Lab1 中无需实现替换策略，主要是实现 getPage 方法。在实验中，利用 pid 来查找 BufferPool 中是否有该数据页，如果有直接返回，如果没有就从磁盘中获取，存入 BufferPool 中，若缓存超出范围，则直接抛出异常。

### Exercise4: HeapFile access method

access method 是从按照特定格式组织的 disk 文件中读取数据库数据的方法。

HeapFile 分成多个 HeapPage，每个页面固定字节数组成，包括 header 和多个 slot，每个 slot 中容纳一个 Tuple。每个页面中有一个 bitmap，用来表示第 i 个 slot 是否被使用。实验中除了基本属性还要实现判断某个 slot 是否被使用。这需要计算一个 page 能存多少条 tuple 和 header 占多少字节，具体计算公式 readme 中已经给出，所以总体还是容易实现的。在后续对数据页进行操作的时候，就要利用这个来判断 slot 是否已经储存了一个元组。同时还要实现一个 iterator。HeapPageId 储存 HeapFile 与 HeapPage 的关系。记录 Page 所在的 tableId 和 page 序号。RecordId 用来唯一标识一个 tuple，记录了 tuple 所在的 PageId 和 tuple 序号。

### Exercise4: HeapFile

HeapFile 是 DFile 的一种实现，随机储存在磁盘上。实验中要实现 readPage，因为是随机储存的，所以要计算文件的偏移量，计算出 PageId 后从磁盘中读取数据。还要实现 iterator 来迭代所有 page 中包含的 Tuple，为了实现这个 iterator，我在 DbFileIterator 的基础上构建了一

个内部类 HeapFileIterator。

### Exercise6: Operators

这里的 Operators 是做一个全表扫描,主要基于上一个 exercise 实现的 iterator,实现 SeqScan,其中 SeqScan 给扫描的表支持别名,然后 getTupleDesc 可以修改每个字段的名字,在最前面加上别名。

### A simple query

先创建一个 some\_data\_file.txt 的文件,然后利用 SimpleDb.class 将其编译为 dat 文件。最后再利用 lab 中提供的代码测试 dat 文件。代码将储存在 some\_data\_file.dat 的表添加到目录,然后进行扫描,将 some\_data\_file 的内容输出出来。

测试:

运行 ant test:

```
BUILD SUCCESSFUL
Total time: 5 seconds
```

运行 anr systemtest:

```
BUILD SUCCESSFUL
Total time: 2 seconds
```

测试 dat 文件:

```
Field_Name:field0==>Value:1
Field_Name:field1==>Value:1
Field_Name:field2==>Value:1

Field_Name:field0==>Value:2
Field_Name:field1==>Value:2
Field_Name:field2==>Value:2

Field_Name:field0==>Value:3
Field_Name:field1==>Value:4
Field_Name:field2==>Value:4
```

## 三、不足

实验中有些冗余的代码。同时一开始不太会用 eclipse,是直接用 txt 打开写的,所以有的格式没怎么对齐,不够整洁。以及由于对 docker 不熟悉,最后没使用 docker 测试,是直接在虚拟机上测试的。

## 四、困难

大概前后花了三天时间完成这个 lab1,最开始的难点在于一开始的配置环境,学习 java 的语法,学习 ant 的使用, eclipse 的配置和使用。并且需要弄明白这个实验的流程,以及 database 的结构,要理解别人写好的大致框架。

在写代码的过程中遇到的问题是,实验中有的类创建完了,再用的时候又忘记了有什么属性和函数,得前后查看翻找着写,因此较为浪费时间。再加上对 java 和数据库不熟练,最开始写的时候报错特别多,有语法错误,有逻辑错误,也有版本配置问题,不过经过多次尝试,终于通过了本地测试。