

Describe any design decisions you made.

2.1 Filter and Join

主要的是 hashjoin

hashjoin , 一个表通过 map 存储

另一个表,不断循环, 然后 logn 的复杂度查询 map 中的元素

为了避免每次 map 查询需要时间太长, 我们可以把 第一个表分块 map 这样就会使得 每次 map 查询的时间复杂度降低, 但是也会因此导致需要循环多次表 2

2.2 Aggregates

实现 Group by 句子

2.3 HeapFile Mutability

并且 Heappage 里面标记错误, 应该是 not necessary for lab1 | lab2

主要的是 bufferpool 中的 insertTuple 和 deleteTuple

我们不在只是需要 ,简单的调用 DbFile 中的 接口, 需要处理 dirtyPage

2.4 Insertion and deletion

```
while (_child.hasNext()) {
    bufferPool.insertTuple(_tid, _tableId, _child.next());
    numTuples++;
}
_active = false;
return newResultTuple(numTuples);
```

2.5 2.6 Query Parser and Contest

三次结果为 0.55 3.94 6.21

```
5 rows.
Transaction 0 committed.
-----
0.55 seconds
```

```
11 rows.
Transaction 1 committed.
-----
3.94 seconds
```

```
Transaction 1 committed.
-----
6.21 seconds
```

Discuss and justify any changes you made to the API.

添加一些必要的 private 成员

Describe any missing or incomplete elements of your code.

缺少一些非法性检测,需要保证,使用者完全按照接口使用.

Describe how long you spent on the lab, and whether there was anything you found particularly difficult or confusing.

花费 大概 30 个小时

突然发现自己 lab1 的 seqscan 写的存在问题,但是 lab1 的 system 是可以通过的.

Permissions 并没有在实验中指出!!!!