1. Environment
- Ubuntu 22.04
- JDK 1.8
- Ant 1.10.12

2. Experimental Procedure

Exercise 1: Filter and Join

The implementations of `Predicate.java`, `JoinPredicate.java`, and `Filter.java` are quite straightforward. The operators use `DbIterators` as their basic iterators to scan tuples and return those that meet specific conditions or are generated by a specific algorithm. `Filter` implements the return of tuples that satisfy filtering conditions, with `Predicate` being part of its constructor. `Join` implements the natural join of two elements, with `JoinPredicate` being part of its constructor. The method of join implemented is a nested loop. Both operations have corresponding predicate helper classes, `Predicate` and `JoinPredicate`, to help compare whether tuple fields meet conditions.

Exercise 2: Aggregates

The code is simple. The task is to perform SQL's grouping (GROUP BY) and aggregation (aggregator) operations.

Exercise 3: HeapFile Mutability

In `HeapPage`, `HeapFile`, and `BufferPool`, implement the functionalities of `deleteTuple` and `insertTuple`. The final modifications are made in the `HeapPage` header, changing the occupancy status of slots. When deleting a tuple, no tuples are actually removed; instead, their slots are marked as unused. For tuple insertion, the first unused slot is scanned, and the new tuple is placed into it.

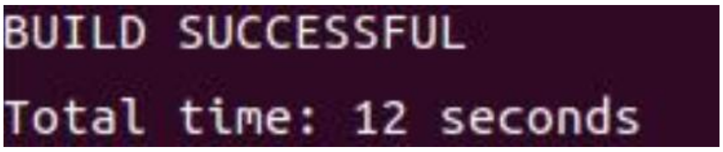
An important point to note is that pages modified should be promptly marked as dirty pages.

Exercise 4: Insertion and Deletion

More about implementing `insertTuple()` and `deleteTuple()`.

3.Run Result:

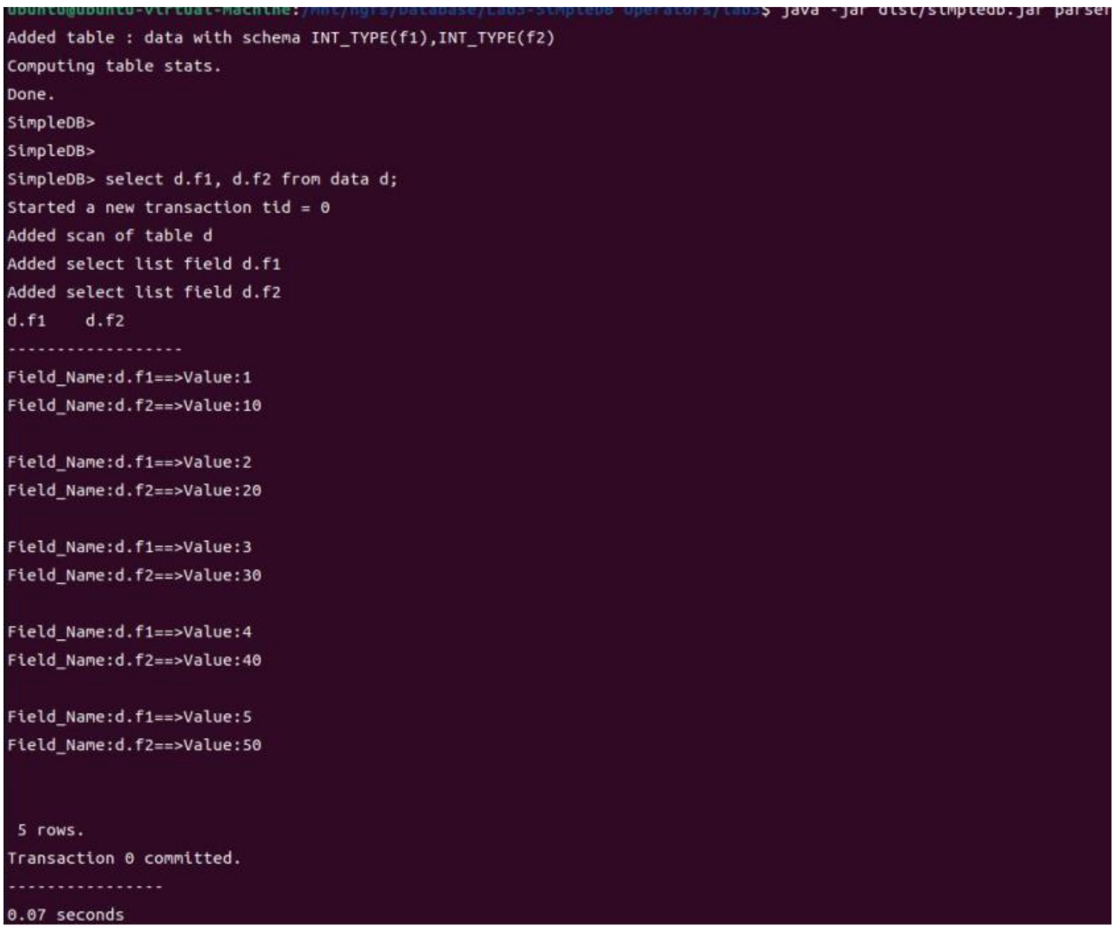
Ant test



```
BUILD SUCCESSFUL

Total time: 12 seconds
```

Query walkthrough



```
ubuntu@ubuntu-virtual-machine:/mnt/hgfs/Database/Lab3-SimpleDB Operators/lab3$ java -jar dist/simpledb.jar parser
Added table : data with schema INT_TYPE(f1),INT_TYPE(f2)
Computing table stats.
Done.
SimpleDB>
SimpleDB>
SimpleDB> select d.f1, d.f2 from data d;
Started a new transaction tid = 0
Added scan of table d
Added select list field d.f1
Added select list field d.f2
d.f1    d.f2
-----------------
Field_Name:d.f1==>Value:1
Field_Name:d.f2==>Value:10

Field_Name:d.f1==>Value:2
Field_Name:d.f2==>Value:20

Field_Name:d.f1==>Value:3
Field_Name:d.f2==>Value:30

Field_Name:d.f1==>Value:4
Field_Name:d.f2==>Value:40

Field_Name:d.f1==>Value:5
Field_Name:d.f2==>Value:50



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

4.Time

I used almost half a week to complete this experiment. And I think this task is easier because it is not that abstract.