Lab 4 **Hanjing Wang(516030910015)**
Database System ACM Class, Zhiyuan College, SJTU
Prof. **Feifei Li** Due Date: June 2, 2019

Design decision

simpledb.IntHistogram

Here in simpledb.IntHistogram, the main design decision I made was dividing the range $[min, max]$ into buckets with integer width. The pros of integer-width buckets are straightforward. First, it reflects the certain nature of the data, which, in other words, means all the values involved are integers. Second, if we set the buckets to have positive integer width, then the following correctness condition is statisfied: the selectivity contribution of a single buckets can never be larger than $height/ntups$. However, there are disadvantages of using a integer-width setting. The most significant one is inconsistency in width, which means the bucket with the largest index probably has different width with the other buckets. It leads to some special decisions in boundary conditions.

simpledb.TableStats

In the construction method of TableStats, I decided to scan the file twice. In the precomputing phase, programs can get the tuple descriptor of the file and therefore get all the types of each fields. In the first scan, programs get min and max values of every integer fields. And then in the second scan, all the tuples are added into the histograms.

simpledb.joinOptimizer

In the estimateJoinCost method, I added a item depended on the output cardinality to for the penalty of range query. Otherwise it will be difficult to meet the request that cost of range query should be larger than that of non-primary key equality join.

Difficulties

I spent around 40 hours in this lab. The major difficulties lie in the implementation of joinOptimizer, which took me a long time to fully understand the framework design of the optimizer.

And I encountered an unusual minor error in testing queries listed in lab3. At last the reason came out to be that a "SELECT" operation on a single table without "AND" statement was considered as a join operation. But obviously, optimizer could not provide any plan for this operation and therefore triggered a NullPtrException when the iterator over the optimized plan was called. The debugging took a long time since the bash environment could only provide few information for analysis.