SimpleDB

Lab3

URDSHALS Amalie AVALLE Dario

Design Decisions

For most classes, we stuck to the delivery: consequently, we only report the significant decisions. Moreover, we did not change the API and we used the provided files from the previous lab.

Join

Since the $HashEquiJoin$ class was already implemented, we called it in the case of equi join.

IntegerAggregator and StringAggregator

The *IntegerAggregator* class has three LinkedHashMap as attributes. The first map (*primaryMap*) is always used and its entries are $\langle$ group by key, aggregated value $\rangle$. The aggregated value is computed during the *mergeTupleIntoGroup* method.
In the case of an average aggregation, the aggregated value of the primary map cannot be the average, so in the *primaryMap* we save the cumulative sum, in a second map (*secondaryMap*) we save the count of values, and in a third map (*avgMap*) we save the temporary average. *StringAggregation*'s implementation is the same, but only the *primaryMap*.

Eviction policy

For what concerns the eviction policy, we stuck to the one already implemented in the inner class $PageBufferPool$ of $BufferPool$: this policy evicts the first page stored in the buffer pool when this is full.

Difficulties encountered

We feel we have spent an appropriate time on the assignment given its size. It is extensive and requires some work. We have faced a few challenges since the task is so big, and all the parts take some time to read up on. Still, we have kept a steady pace with the assignment and haven't faced too many difficulties. Once we understood how it all fits together and visualized the SimpleDB and its components, it was an ok task.