REPORT Project 1

Sara Djambazovska

In the first part of the Database Systems Project, I implemented the six basic operators, in a volcano-style tuple-at-a-time operators: ***Aggregate, Filter, Join, Project, Scan*** and ***Sort*** in their respective *.scala* classes in the package ch.epfl.dias.cs422.rel.early.volcano. Only the Scan implementation is modified from Project 0, to read RLE compressed columns and output uncompressed tuples.

The implementation of Decode is quite straight forward, keeping track of the length of the current tuple and calling next only once it is 0.

Reconstruct was implemented by basically merging the left and right tuples, knowing that only their intersection (in terms of startVID and length) should be taken.

The RLE Operators were implemented similarly as the tuple at a time, with the particularity that in RLE Join the resulting tuple's length is the product of the right and left tuple's lengths, and the length of an RLEAggregate resulting tuple is always 1, taking into account the repeating tuples in the getArgument call.

For the next part of implementing the query optimizers, the idea was similar as in the provided query optimizers in the helpers package.

For the execution models, the implementation was similar as for tuple at a time, as I first transpose the (homogeneous) columns and work on the tuples and then transpose them back to return them in column form.

Hash Join was used in all Join implementations