Query Processing: Sort Operator

These are operations of query manager, which is yet to be introduced. So we have no constructor here.

sort(smarterator<tuple>, array<attrId>, numberOfBlocks)
Returns sorted smarterator of tuples.

To be consistent with selection operator, this function takes an iterator of tuples. array<attrId> is the array of attributes we want to sort by, sorting in order, from array(0) to array(n). Number of blocks is the main memory buffer size necessary to perform sorting (see “runs” in sort-merge algorithm).

This function is used to sort tuples after other queries, so we don’t have to access relations on disk. For example SELECT * FROM A ORDER BY name ASC; We work directly with result of selection.

Current implementation of the algorithm is such that it always tries to perform sort-merge (even if there is enough size in memory for all tuples) and performs quick sort in each run. So in case we have enough memory for all tuples, it runs once.

sort(rId, array<attrId>, numberOfBlocks)
Returns sorted smarterator of tuples.

Totally same as first function, but takes relation id instead of in-memory-smarterator.