CS 4321/5321 Project 3 (Preview)

Fall 2016

Checkpoint due Monday October 17th, 11:59 pm
Project due Tuesday November 1st, 11:59 pm

This project counts for 24% of your grade, of which 6% is for the checkpoint and 18% for the main submission.

1 Goals and important points

In Project 2, you developed a lot of functionality. However, the priority was end-to-end evaluation of SQL rather than efficiency. So, you used line-at-a-time I/O. You also used a naïve implementation for joins (tuple-nested loop join, TNLJ) and for sorting (in-memory sort). Your sort implementation was particularly problematic because it kept *unbounded state* – the amount of memory required by the operator depended on the size of the input rather than being bounded by a constant.

For Project 3, you will address the above shortcomings:

- you will move from using line-at-a-time I/O to faster page-at-a-time I/O
- you will refactor your code to support multiple different physical implementations for each relational algebra operator
- you will implement Block Nested Loop Join (BNLJ), External Sort and Sort Merge Join (SMJ)
- you will ensure that you have at least one implementation for each operator that does not keep unbounded state
- you will do some performance benchmarking of your join implementations against each other

You will still support the same subset of SQL as for Project 2, and you should still construct the join tree in the same manner as in Project 2, following the query's `FROM` clause.

This is a challenging Project, including significant refactoring and nontrivial algorithms to implement. You have a substantial time window to do it, but this window also includes Fall Break and the 4320 prelim. Thus you need to budget your time wisely.

To encourage you to get started early, we are requiring a *checkpoint* submission on October 17th The checkpoint requirements and detailed instructions for all of Project 3 will be available later.