CS 6530 Database Systems

Project 4: SimpleDB Query Optimization

Team 15 - Tanvi Gangadhar(u1205740), Greeshma Mahadeva Prasad (u1141804)

DECISION DESIGN

Statistics Estimation

- Filter Selectivity -
  Implemented IntHistogram by applying the equi-width histogram bucket based technique described in the project description using fixed number of buckets as provided. We are maintaining a bucket array that is utilized to estimate selectivity for the *equals, like*, using the expression (h / w) / ntups where the width (range of values) of the bucket is w, the height (number of tuples) is h, and the number of tuples in the table is ntups. Similarly, the selectivity of a range expression is also determined using the formulae as described in the project description.
  For TableStats, we are maintaining 4 HashMaps for min values, max values, integer histogram and string histogram. We are scanning through each tuple to populate these maps and also have modified the constructor as required and implemented the mentioned methods.
- Join Cardinality -
  estimateJoinCost and estimateJoinCardinality have been implemented assuming the join is a nested loop join.

Join Ordering

- For JoinOptimizer, the pseudo code provided has been followed looping through subset sizes, subsets, and sub-plans of subsets, calling computeCostAndCardOfSubplan and building a PlanCache object that stores the minimal-cost way to perform each subset join. Implementation of other helper methods in the class provided hints for this implementation.

API CHANGES

We have not made any changes to the provided API. We did not have to implement any new classes for this project.

MISSING/INCOMPLETE ELEMENTS

All the unit and system test cases passed, we believe we have implemented everything needed as per project 4 requirements.

TIME SPENT

Similar to all previous projects, we pair programmed most of part of the project and together we spent about 20 hours over a few days. The breakup is as shown below,

- Understanding how overall plan cost and join cost is estimated and the overall control flow of simpleDB - around 3 hours
- Statistics Estimation- 7 hours
- Join Ordering - 10 hours

DIFFICULTIES ENCOUNTERED

Understanding the various classes involved with JoinOptimizer such as PlanCache, CostCard, LogicalJoinNode, LogicalPlan etc