

**CS 123 Project Proposal: NanoDB+**

Angela Gong, Mike Qian, Kalpana Suraesh

**Overall Goal and Features**

Implement grouping and aggregation in NanoDB, including grouping on columns, computing aggregates on columns, and expressions involving aggregates

**Implementation Details**

- We will be using NanoDB, a database written in Java using code from Donnie
- Extending NanoDB to support grouping and aggregate functions
- Programming language: Java
- Version control system: git
- Repository stored on CS cluster (perhaps ~/cs/courses/cs123/sp2012/nanodbpplus/)

Outline	People	Time per person
Proposal	Everyone	Week 1: 4 hours
Set up repository	Everyone	Week 1: 1 hour
Write test cases - Make sure tests are thorough	Mike	Week 1: 3 hours
Understand parser and add functionality for GROUP BY/aggregates - Should parse GROUP BY, MIN, MAX, AVG, COUNT, SUM, STDDEV, VARIANCE, DISTINCT, etc. - Also collect grouping column tokens and add to a Collection	Kalpana, Angela	Week 1-2: 6 hours
Create nodes for grouping and prepare framework for aggregation	Angela, Mike	Week 2: 10 hours
Implement GROUP BY for one column - Update planner to incorporate GROUP BY nodes (Angela) - Figure out how to rearrange tuple order (Kalpana) - Implement grouping with hashing (Everyone) - Implement grouping with sorting (Everyone)	Everyone	Week 3: 5 hours (Milestone 1)
Implement GROUP BY for multiple columns - Expand upon grouping for one column - Add recursive sorting algorithm to sort multiple columns	Everyone	Week 3: 4 hours
Fix project nodes if they don't work with grouping - Determine the order in which to project, using either switch or aliases (Angela, Mike)	Everyone	Week 3-4: 7 hours
Debug GROUP BY, make sure all grouping tests work - Make sure database still runs other commands normally (Mike)	Everyone	Week 4: 10 hours

Outline	People	Time per person
Have parser work with aggregate nodes and planner <ul style="list-style-type: none"> <li>- Add clauses corresponding to aggregate functions to a Collection (Kalpana, Mike)</li> <li>- Update planner to include aggregate nodes (Angela)</li> </ul>	Everyone	Week 5: 10 hours
Implement aggregate expressions without grouping <ul style="list-style-type: none"> <li>- Implement MIN, MAX, SUM, COUNT, AVG, STDDEV, VARIANCE</li> <li>- Implement DISTINCT (Angela)</li> <li>- Fix any renaming issues (Kalpana)</li> </ul>	Everyone	Week 6: 8 hours (Milestone 2)
Implement aggregate expressions with grouping <ul style="list-style-type: none"> <li>- Make sure renaming works correctly before and after grouping</li> <li>- Make sure projection is done correctly (Kalpana)</li> </ul>	Everyone	Week 6-7: 10 hours
Debug aggregation, make sure all tests work now <ul style="list-style-type: none"> <li>- Run with all sorts of commands (Kalpana)</li> <li>- Make sure tests are thorough (Angela)</li> <li>- Make sure database still runs other commands normally (Mike)</li> </ul>	Everyone	Week 8: 12 hours
More debugging and final fixes, clean up code, finish and polish documentation	Everyone	Week 9: 5 hours (Milestone 3)
Final report and presentation	Everyone	Week 9-10: 6 hours