© 2011 Sudarshan S. Chawathe

These questions complement continuing detailed study of the packet-grouping query from the *AQuery* paper[1] and related performance discussions based on Graefe's survey[2] and experiments with PostgreSQL.

The focus of this exercise is studying the *best ways to compute* the desired query results using the available tools, specifically PostgreSQL installations of different versions, windowing functions, and PL/pgSQL and host language code as needed. The *primary requirement* is that the overall computation time be competitive with, if not better than, hand-coded solutions.

1. List the members of your group below. Underline your name.

2. Provide an *efficient* SQL expression of the *packet-grouping* query using the window-function features of PostgreSQL 9.0.

   Hint: Consider the PostgreSQL select-clause syntax similar to the SQL:1999 syntax from the AQuery paper:

```
...  aggrfunc OVER (PARTITION BY a1, a2, ...
                    ORDER BY a3, a4, ...
                    ROWS N PRECEEDING)...
```

[1] Alberto Lerner and Dennis Shasha, "AQuery: Query Language for Ordered Data, Optimization Techniques, and Experiments," in *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB)* (Berlin, Germany, 2003).

[2] Goetz Graefe, "Query evaluation techniques for large databases," *ACM Computing Surveys* 25/2 (1993).

3. Repeat Question 2 for PostgreSQL 8.4, which supports the `rows N preceding` syntax only with `N=unbounded`.

   Hint: Consider the `lag` window function with `lag`(a, N, d) giving the value of attribute a from the N'th previous tuple, or d if there is no such tuple.

4. Repeat Question 2 for PostgreSQL 7.0, which does not support window functions.