Similarity Group-by

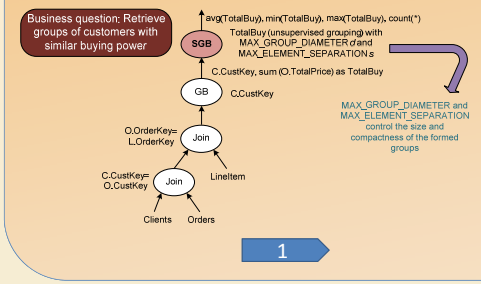
Yasin N. Silva[1], Walid G. Aref[1], and Mohamed H. Ali[2]
[1]Purdue University - [2] Microsoft Corporation

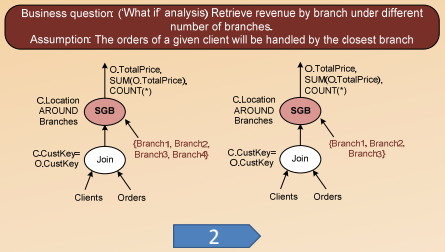
ICDS Indiana Center for Database Systems

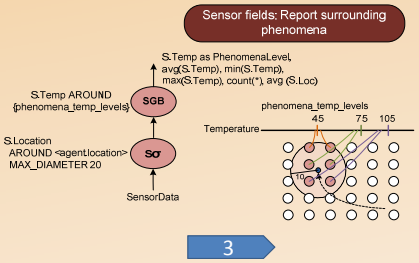
PURDUE UNIVERSITY

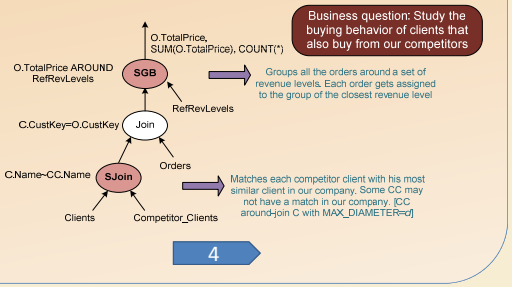
Use Case Scenarios

- Grouping features are extensively used in OLTP, OLAP, and decision support systems.
- Many applications scenarios can benefit from queries that take advantage of similarities in the data (biology, sensor networks, business)









Similarity Group-by (SGB)

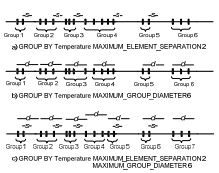
- We propose to extend the standard grouping operators to group similar or approximate values
- The main goal of SGB is to generate more meaningful and useful similarity-based groupings than those of the regular group-by while maintaining:
  - Low running time
  - Good scalability properties
  - Efficient integration with the query processing engine

SGB: Three Instances

Unsupervised SGB

```
SELECT select_expr, ...
FROM table_references WHERE where_condition
GROUP BY col_name
    [MAXIMUM_ELEMENT_SEPARATION s]
    [MAXIMUM_GROUP_DIAMETER d], ...
```

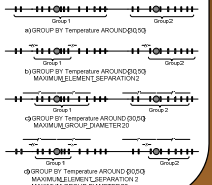
- No extra data provided to guide the process
- Clauses control group size and group compactness



Supervised Similarity Group Around

```
SELECT select_expr, ...
FROM table_references WHERE where_condition
GROUP BY col_name AROUND central-points
    [MAXIMUM_GROUP_DIAMETER 2r]
    [MAXIMUM_ELEMENT_SEPARATION s], ...
```

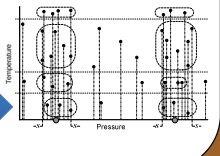
- Groups tuples *around* a set of guiding points
- Each tuple is assigned to the group of its closest central point.
- Clauses control group size and group compactness



Supervised Similarity Group with Delimiters

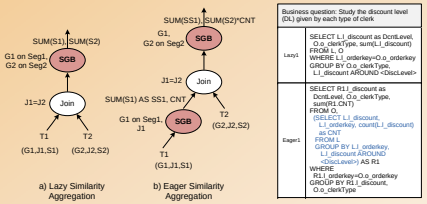
```
SELECT Avg(Temperature), Avg(Pressure)
FROM SensorsReadings
GROUP BY
Temperature DELIMITED BY (SELECT Value FROM Thresholds),
Pressure AROUND {30,50} MAXIMUM_GROUP_SEPARATION 3
```

- Forms groups based on a set of *delimiting* points
- Several similarity grouping strategies in the same SQL statement
- Each grouping attribute can use a different strategy



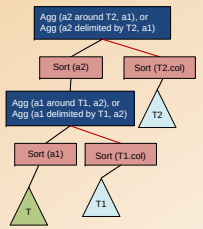
Optimization

- Materialized views can be used to answer similarity queries (details in paper)
- Eager/Lazy similarity aggregation (Main theorem)



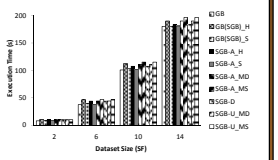
Implementation (PostgreSQL)

- The Parser
  - Extended the grammar rules and parse tree structure
- The Planer/Optimizer
  - Made use of the RHS input plan tree of aggregation nodes
  - Each aggregation node processes 1 SGA and 1 or more GAs
  - SGAs can be ordered to reduce number of flowing tuples
- The executor
  - Hash-based approach
  - Single plane sweep approach

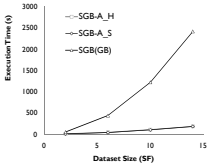


Performance Evaluation (TPC-H)

Performance while increasing dataset size

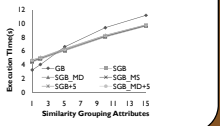


Performance of generating similarity groups with GB vs. SGB



Not all the similarity grouping strategies can be obtained using only regular group-by

Performance while increasing SGAs



Performance of complex queries