# Exploiting Similarity-aware Grouping in Decision Support Systems*

Yasin N. Silva
Dept. of Computer Science
Purdue University
West Lafayette, IN, USA
ysilva@cs.purdue.edu

Muhammad U. Arshad
Dept. of Electrical & Computer Engg.
Purdue University
West Lafayette, IN, USA
marshad@purdue.edu

Walid G. Aref
Dept. of Computer Science
Purdue University
West Lafayette, IN, USA
aref@cs.purdue.edu

## ABSTRACT

Decision Support Systems (DSS) are information systems that support decision making processes. In many scenarios these systems are built on top of data managed by DBMSs and make extensive use of its underlying grouping and aggregation capabilities, i.e., *Group-by* operation. Unfortunately, the standard grouping operator has the inherent limitation of being based only on equality, i.e., all the tuples in a group share the same values of the grouping attributes. Similarity-based Group-by (SGB) has been recently proposed as an extension aimed to overcome this limitation. SGB allows fast formation of groups with similar objects under different grouping strategies and the pipelining of results for further processing. This demonstration presents how SGB can be effectively used to build useful DSSs. The presented DSS has been built around the data model and queries of the TPC-H benchmark intending to be representative of complex business analysis applications. The system provides intuitive dashboards that exploit similarity aggregation queries to analyze: (1) customer clustering, (2) profit and revenue, (3) marketing campaigns, and (4) discounts. The presented DSS runs on top of PostgreSQL whose query engine is extended with similarity grouping operators.

## 1. INTRODUCTION

In general terms, a Decision Support System (DSS) can be defined as a tool that supports decision making tasks. Different taxonomies of DSS have been proposed, among them [5] recognizes the following DSS types: Database-oriented DSS, Text-oriented DSS, Spreadsheet-oriented DSS, Rule-oriented DSS, Solver-oriented DSS, and Compound DSS. Database-oriented DSSs make use of data managed by a DBMS and make extensive use of its underlying grouping and aggregation capabilities. The most commonly used grouping database capability is provided through the *Group-by* operation. This operation has simple semantics and relatively good execution time and scalability properties. However, it also has the key limitation of being based only on equality, i.e., all the tuples in a group share the same values of the grouping attributes.

| U-SGB: Unsupervised Similarity Group-by |
|---|
| SELECT select_expr, ...<br>FROM table_references WHERE where_condition<br>GROUP BY col_name<br>[MAXIMUM_ELEMENT_SEPARATION s]<br>[MAXIMUM_GROUP_DIAMETER d], ... |
| **SGB-A: Supervised Similarity Group Around** |
| SELECT select_expr, ...<br>FROM table_references WHERE where_condition<br>GROUP BY col_name AROUND central-points<br>[MAXIMUM_GROUP_DIAMETER 2r]<br>[MAXIMUM_ELEMENT_SEPARATION s], ... |
| **SGB-D: Supervised SGB using Delimiters** |
| SELECT select_expr, …<br>FROM table_references WHERE where_condition<br>GROUP BY col_name DELIMITED BY limit-points |

**Figure 1. Supported similarity Group-by syntax**

Similarity-based grouping has been studied recently to overcome the limitations of the regular Group-by [1, 2, 3]. In particular, [3] proposes Similarity Group-by (SGB), an extension of its non-similarity counterpart that allows the formation of groups of similar objects under different grouping strategies. SGB is implemented as a native operator inside the query engine in contrast to other techniques to support clustering or data mining models outside of the engine. This implementation at the engine level allows: (1) full integration with the query processor, (2) the pipelining of results for further processing, and (3) the use of optimization techniques, e.g., query transformations or using materialized views, to optimize complex queries. In contrast to expensive clustering techniques, SGB operators have been shown to have very low execution time and good scalability properties, i.e., at most only 25% more expensive than the regular Group-by.

In this demonstration, we present an implementation of SGB operators supporting different grouping strategies inside the PostgreSQL query engine. We also show how similarity-based grouping can be effectively used to build very useful DSSs. We present a DSS built around the data model and queries of the TPC-H benchmark [4] intending to be representative of complex

* This work was partially supported by NSF Grant IIS-0811954 and by NIH NIGMS U24 GM077905 for the EcoliHub project.


EDBT'09, March 24-26, 2009, Saint Petersburg, Russia.

business analysis applications. The TPC-H data model and queries have been extended to allow for similarity-based queries. The user interface is composed of several analysis dashboards. Each of them supports the analysis and decision making process associated to a specific business question. Dashboards make use of similarity grouping queries to analyze: (1) customer clustering, (2) profit and revenue, (3) marketing campaigns, and (4) discounts.

The remaining part of this paper is organized as follows. Section 2 presents briefly the implemented SGB strategies. The main components of our decision support system are described in Section 3. Section 4 discusses in some detail how the similarity aggregation features are used by the implemented analysis dashboards. Section 5 presents the conclusions and future work.

## 2. SIMILARITY-AWARE GROUPING

We implement three instances of the similarity grouping operator as described in [3]: (1) U-SGB: Unsupervised Similarity Group-by, (2) SGB-A: Supervised Similarity Group Around, and (3) SGB-D: Supervised SGB using Delimiters. U-SGB produces similarity groups based only on the specification of group properties, e.g., compactness and size. S-GBA identifies the groups formed around certain central points of interest and limits their extent based on group properties as in U-SGB. SGB-D creates groups based on a set of delimiting points. It is important to notice that both central and delimiting points can be specified using a generic query. Consequently, these points can dynamically change over time. Figure 1 presents a summary of the supported syntax and Figure 2 shows some examples of the implemented operators. Different grouping strategies can be combined in a single similarity aggregation query. In this case, each aggregation attribute can make use of a different similarity grouping strategy.

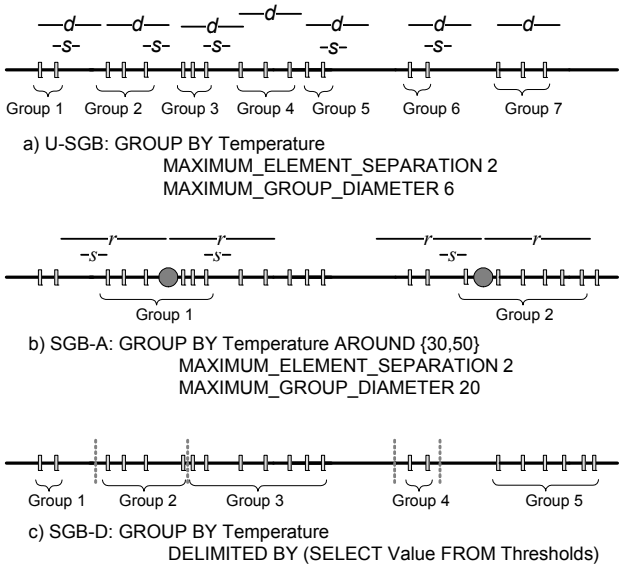


**Figure 2. Examples of similarity-aware grouping**

## 3. DSS ARCHITECTURE

The architecture of the decision support system is presented in Figure 3. At the top level, the analysis dashboards constitute the visual tools that decision makers use in their decision making activities. Dashboards make extensive use of similarity aggregation queries to (1) identify and analyze the profit, revenue, effect of promotions, and customer satisfaction associated to groups of similar clients, providers, and products; and (2) analyze sales, revenues, and customer satisfaction around certain important events, e.g., holidays or marketing campaigns. The similarity aggregation queries make use of one or several similarity grouping operators. The operators for the different supported strategies, i.e., U-SGB, SGB-A, and SGB-D, are implemented in the query engine of PostgreSQL. The data model is based on the one used in the TPC-H benchmark. Figure 4 presents the TPC-H tables, and additional reference points tables, i.e., central and delimiting points, used in our system.

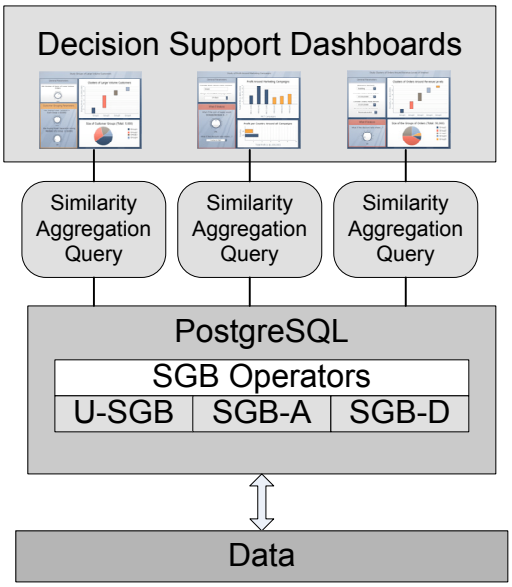


**Figure 3. Architecture of the decision support system**

| TPC-H Tables | |
|---|---|
| Part(P), Supplier(S), PartSupp(PS), Customer(C), Orders(O), LineItem(L), Nation(N) | |
| **Reference Points Tables** | |
| **RevLevels**: | 10 order revenue levels. {10000,50000,…,370000} |
| **MktCmpDates**: | Marketing campaign dates. Random in the range of O_orderdate |

**Figure 4. Used tables and additional attributes**

## 4. USING SIMILARITY-AWARE GROUPING IN DSSs

There are many ways in which similarity-based grouping can be used to build effective decision support tools. This section describes the use of SGB to support more effective analysis in some of the dashboards of our DSS.

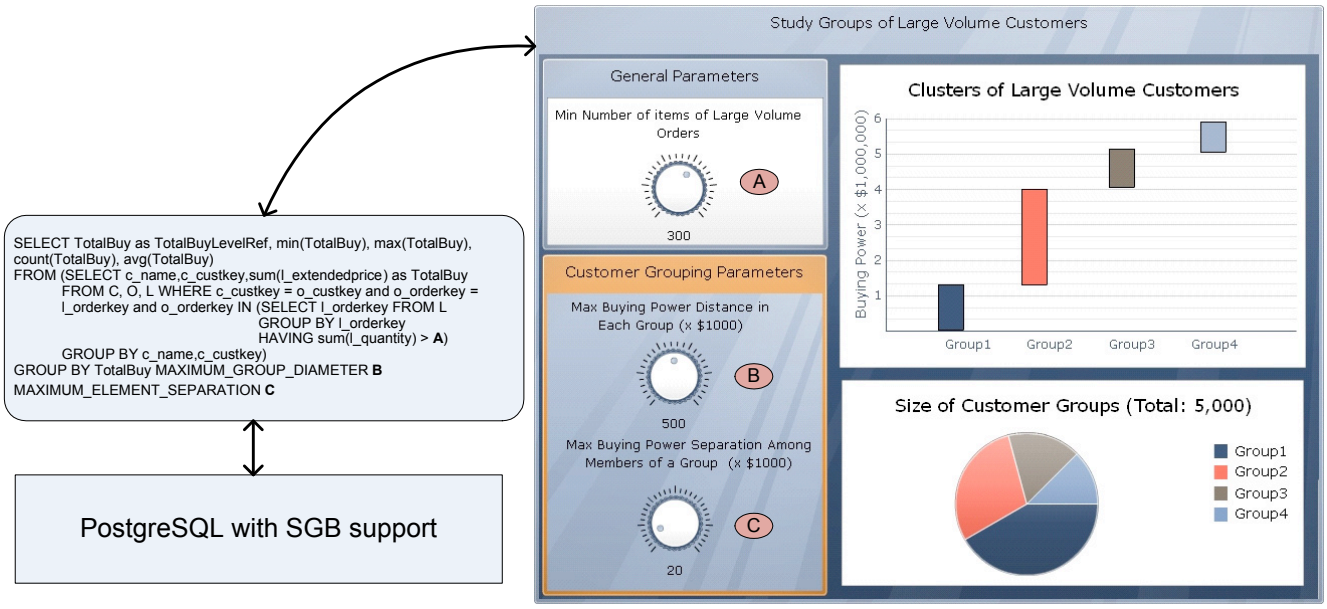


**Figure 5. Dashboard 1 - Studying groups of large-volume customers with similar buying power**

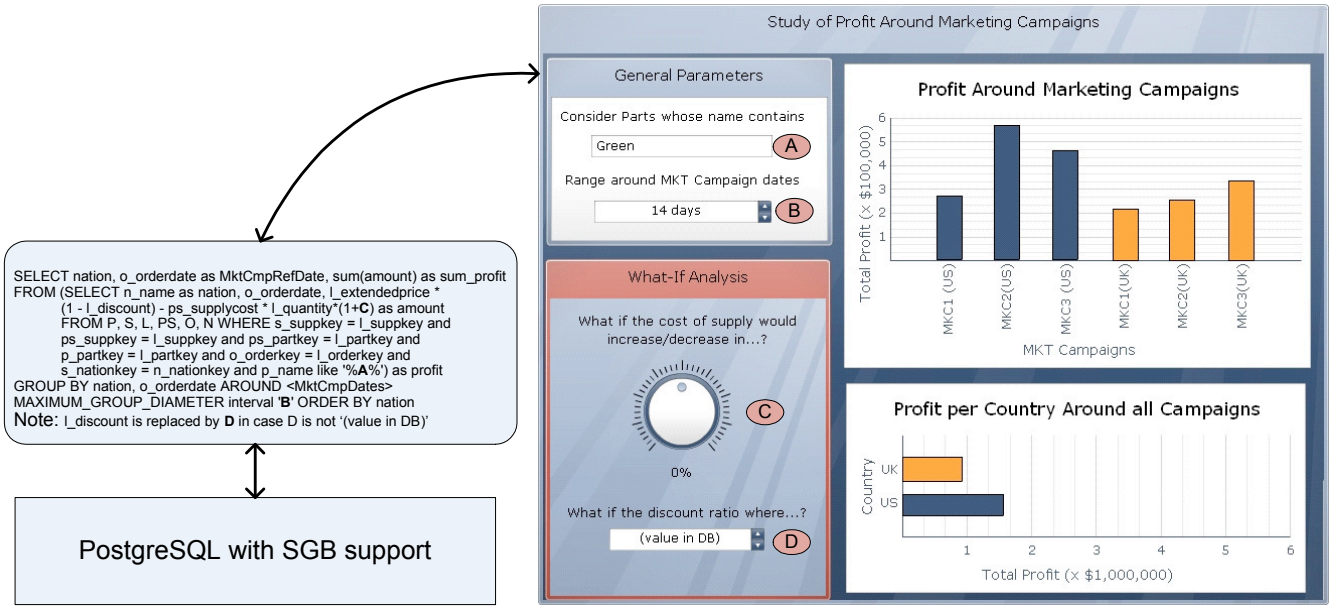


**Figure 6. Dashboard 2 - Studying profit of a line of parts around marketing campaigns**

## 4.1 Dashboard 1: Studying groups of large-volume customers with similar buying power

The user interface of this dashboard is shown in Figure 5. This dashboard allows the study of groups of customers with similar buying power, i.e., total revenue due to large volume orders. The dashboard allows the dynamic specification of the minimum number of items for an order to be considered as a "large volume" order, and properties that describe the similarity groups to be formed: group-size (maximum buying power distance in each group), and compactness (maximum buying power separation among members of a group). The dynamic values of these operators are used to build the similarity aggregation query shown in Figure 5. The dashboard is updated after any change on the parameters' values.

## 4.2 Dashboard 2: Studying profit of a line of parts around marketing campaigns

This dashboard is shown in Figure 6. It supports the study of changes on profit around marketing campaigns. The supported generic parameters are: (1) keywords for the product name, and (2) range around the marketing campaign dates. Furthermore, this dashboard supports the study of what-if scenarios. Specifically, it allows generating the profits if the cost of supplies or the discount ratios would have been different. Figure 6 also presents

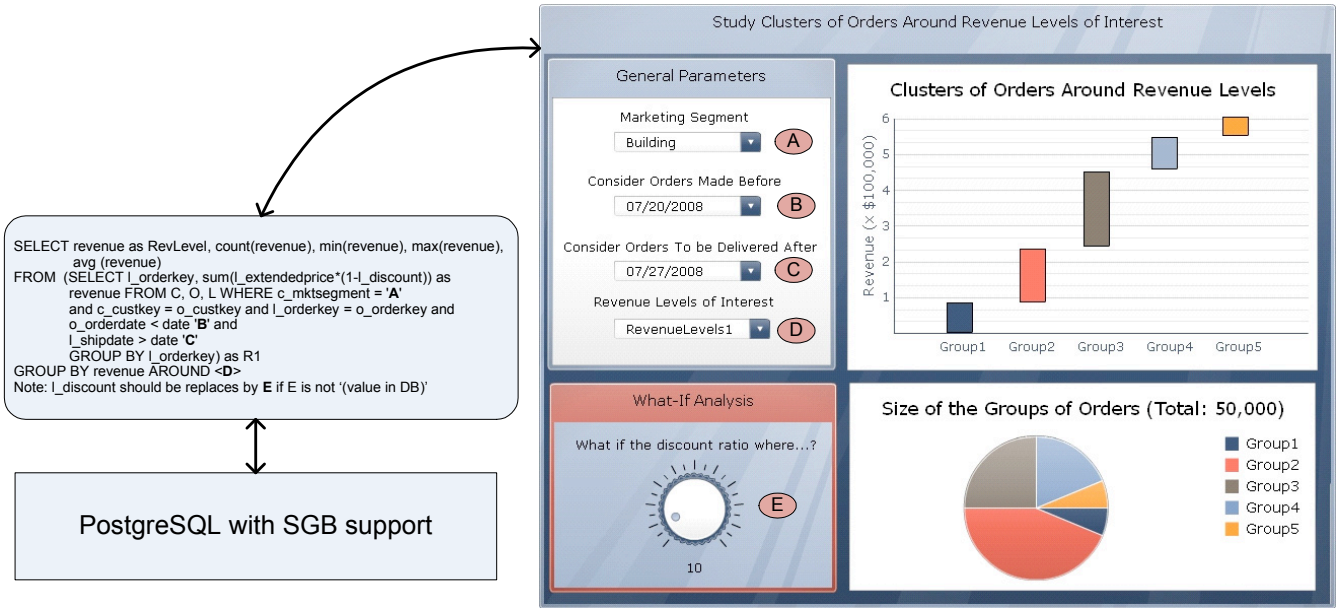


**Figure 7. Dashboard 3 - Studying groups of orders around revenue levels of interest**

the similarity aggregation query used by this dashboard.

## 4.3 Dashboard 3: Studying groups of orders around revenue levels of interest

The interface of this dashboard is shown in Figure 7. This dashboard supports the study of groups of similar orders around certain revenue levels of interest. The supported generic parameters can be used to dynamically specify: (1) the marketing segment of customers, (2) maximum date of order, (3) minimum date of delivery, and (4) the list of revenue levels of interest which could be entered directly as a sequence of values or could be the obtained from a database table. In addition, this dashboard allows changing the discount ratio to study the effect that this would have in the formed clusters and their associated revenue. The similarity aggregation query used by this dashboard is presented in Figure 7.

# 5. CONCLUSIONS AND FUTURE WORK

This demonstration presents an implementation of Similarity Group-by operators (U-SGB, SGB-A, and SGB-D) within an open source database management system, PostgreSQL. We show how similarity-based grouping can be exploited to build more useful DSSs. We present the implementation of a DSS that intends to be representative of complex business analysis applications. This DSS is composed of several analysis dashboards that aim to supports the decision making process associated to different business questions. This paper describes the details of three dashboards that support: (1) the study of groups of large-volume customers with similar buying power, (2) the study of profit of lines of parts around marketing campaigns, and (3) the study of clusters of orders around revenue levels of interest. These dashboards support the dynamic modification of parameters allowing decision makers to focus on the subset of data of interest, and the evaluation of what-if scenarios. The support of dynamic modification of parameters is possible due to the good execution time and scalability properties of SGB, i.e., only 25% more expensive than Group-by.

Some paths for future work include the integration of similarity join techniques and similarity selection operators within the PostgreSQL query engine. Decision support systems are only one scenario in which SGB can be effectively used to support queries that could be too complex, or too expensive to answer using only the non similarity-aware operators. The experimental study of similarity-aware query operators in other areas, e.g., sensor networks, biology, medicine and psychology, is another task for future work. Also, in the area of business and data warehouses, SGB ideas could be extended to the case of the extensively used CUBE and ROLAP operators.

# 6. REFERENCES

[1] E. Schallehn, K. –U. Sattler, and G. Saake, "Efficient Similarity-based Operations for Data Integration," *Data & Knowledge Engineering*, vol. 48, no. 3, pp. 361-387, 2004.

[2] C. Li, M. Wang, L. Lim, H. Wang, and K. C. –C. Chang, "Supporting Ranking and Clustering as Generalized Order-by and Group-by," In *Proc. SIGMOD' 07*, pp. 127-138, 2007.

[3] Yasin N. Silva, Walid G. Aref, and Mohamed H. Ali, "Similarity Group-by," In *Proc. ICDE' 09*, 2009, (To Appear).

[4] TPC-H Version 2.6.1 in PDF Form. [Online]. http://www.tpc.org/tpch/default.asp

[5] C. W. Holsapple, A. B. Whinston, "*Decision Support Systems: A Knowledge-Based Approach*," 1st Edition, West Group Publishing, 1996.