

Extensible and Similarity-based Grouping for Data Integration

Eike Schallehn Kai-Uwe Sattler Gunter Saake
Department of Computer Science, University of Magdeburg
P.O. Box 4120, D-39016 Magdeburg, Germany
{eike|kus|saake}@iti.cs.uni-magdeburg.de

Abstract

The general concept of grouping and aggregation appears to be a fitting paradigm for various issues in data integration, but in its common form of equality-based grouping a number of problems remain unsolved. We propose a generic approach to user-defined grouping as part of a SQL extension, allowing for more complex functions, for instance integration of data mining algorithms. Furthermore, we discuss high-level language primitives for common applications.

1. Introduction

Data integration as required in a variety of applications like data warehousing, information system integration etc. makes great demands regarding features to deal with overlapping or otherwise related data. As part of our work on a framework for data fusion we extended the well known concept of grouping and aggregation to deal with more complex relationships hidden in data sets, and to reconcile inconsistent representations. While extensible aggregation is a common concept and part of the SQL standard, we focus on concepts for extensible grouping.

2. Extensible and Similarity-based Grouping

As an example for similarity-based grouping consider the following query, that integrates instances from various sources of bibliographical data and reconciles them based on information about data quality.

```
select pickByQuality(title,quality)
from DBLP union SPRINGER union NCSTRL
group by transitive similarity
on sameText(title) and
sameName(author) or isbn
threshold 0.95
```

In this case group membership is derived from a pairwise similarity measure expressed in the **on**-clause and a the relationship is established along the transitive closure where the similarity exceeds the specified threshold. Similarity is expressed in terms of system- and user-defined functions and their logical combination. The rather specialized language construct described in the previous example can be implemented based on the more general concept of context-aware grouping used in the following query.

```
select pickByQuality(title,quality)
from DBLP union SPRINGER union NCSTRL
group by context
samePublication(title,author,isbn)
```

The user-defined grouping function `samePublication` is implemented based on a system-defined interface that consists of simple iteration methods for the grouping stage, as well as for the group and tuple access during aggregation. The latter is described in more detail in [1].

3. Implementation and Optimization

An implementation is part of our own federated query engine, but we are currently working on integrating the approach with commercial system based their extensibility interfaces, e.g. table functions in Oracle. Because efficient implementations of user-defined grouping functions can become very complex, we propose to provide this functionality as packages for certain integration scenarios.

Another current focus of our work is to support similarity based grouping with according index structures and apply the proposed concepts in specific applications.

References

- [1] E. Schallehn, K. Sattler, and G. Saake. Extensible grouping and aggregation for data reconciliation. In *Proc. 4th Int. Workshop on Engineering Federated Information Systems, EFIS'01, Berlin, Germany, 2001.*