Data Integration using self-maintainable views

Rajarshi Sarkar

Last Updated: May 23, 2014
## Contents

1 Introduction 1

2 Notation and Background 1

3 Self Maintenance for SPJ Views 2
   3.1 Insertion 2
   3.2 Deletion 2
   3.3 Updates 2
1 Introduction

Self maintainable views are views that can be maintained using only the contents of the view and the database modifications but without accessing any of the underlying databases. Many of these databases may be legacy systems, or systems separated for organizational reasons like funding and ownership. Integrating data from such distinct databases is a pressing business need. A common approach for integration is to define an integrated view and then map queries on the integrated view onto queries on the individual systems. This model is not good for applications where response time is critical or for decision support applications with complex queries.

The authors of the paper propose an alternative model good for applications where fast response to queries and high availability are important. The integrated view is materialized and stored in a database. Queries on the view are then answered directly from the stored view. This approach involves the additional storage expense for the materialized integrated view.

2 Notation and Background

It may be expensive or impractical to obtain much more than just a periodic report of local updates from each underlying database. Under such circumstances it becomes crucial that the integrated view be self maintainable or Self-M meaning that view maintenance should be possible without requiring access to any underlying database and without access to any information beyond the view itself and the log of the changes. Self-M can be used in data warehousing environments to:

- Efficiently maintain views.
- Avoid concurrency control problems.

Self maintenance problem is defined as the problem of maintaining a view in response to insertions, deletions or update collectively referred to as modifications using only the view and the set of changes to the referenced relations without access to the full referenced relations. Here are some definitions:

- key(R): It refers to the key attributes of R. For a relation R that has a key r a tuple in R is said to be updated if one or more attributes of the tuple are assigned a value different from its original value. An update to tuple r that results in tuple p is represented as m(r,p).

- Self Maintainability with respect to a Modification: A view V is said to be self-M with respect to a modification type (insertion, deletion, or update) if for all database states, the view can be self maintained in response to a modification of that type to the base relations.

- Exposed Variable: Given a view definition V, a variable, or equivalently, an attribute, A, of a relation used in the view definition is said to be exposed if it is involved in some predicate.
3 Self Maintenance for SPJ Views

The authors restrict themselves in classifying self maintainability with respect to the relation being modified the type of modification and key information. They do not consider the other finer granularity distinctions.

3.1 Insertion

- Insertion Views are self-M for insertions only under very limited circumstances. It is not possible to self maintain an SPJ view joining at least two distinct relations upon an insertion into a component relation. Even a view involving a self join of a relation with itself may not be self-M.

- All SP views are self-M with respect to insertions. A SPJ view defined using self joins over a single relation R is self-M if every join is based on key(R).

- For SP views self maintenance can be done by a selection and a projection on the newly inserted tuples. For a SPJ view with self joins on a key each newly inserted tuple joins only with itself so the view definition is evaluated on only the set of inserted tuples.

3.2 Deletion

- For deleted tuple r in relation R check key(r) satisfies any predicates that equate key attributes to constants in the view definition. Then, look for tuples in the view that have the same values on the remaining key attributes as the deleted tuple. Delete all such tuples from the view.

3.3 Updates

- An update means that a view is self-M with respect to updates only if the view is self-M with respect to both inserts and deletes.

- A view may be self-M with respect to updates even it is not self-M with respect to both insertions and deletions because by modeling an update directly, not as a delete plus an insert we retain the link between the deleted and inserted tuples. Thus often most of the attributes of the new tuple that needs to be inserted into the view can be obtained from the deleted tuple enabling self maintenance.

- A SPJ view V that joins two or more distinct relations is self-M with respect to updates to relation R if and only if either:

  1. The updated attributes are non-exposed and are not distinguished with respect to view V or
  2. The updated attributes are non-exposed with respect to view V and V is self-M with respect to deletions to the relation R.
References
