

PERFORMANCE ANALYSIS OF THE ALGORITHMS FOR THE *OUTER-JOIN* OPERATOR EVALUATION

Gabriela FLOREA
University of Bucharest
e-mail: criogab@yahoo.com

Abstract

The *outer join* operator is important for combining data that resides in several relations that have at least an attribute in common. Often in joining two relations it happens that *join* attributes don't have a matching value. Using different forms of *outer join* operator (*left outer join*, *right outer join* and *full outer join*) all tuples, joinable or not, will be included in the result.

The *outer join* operator is important because it preserve information that would have been lost by using other types of *join*. The most common algorithms for processing *join* operator (*Nested Loops*, *Sort Merge* and *Hash Join* [1]) were used to create specific algorithms for implementing *outer join* operators. What algorithm is efficient in case of *outer join* operator? This article tries to answer that question.