

SCCS/M3937001 : MAJOR : COMPUTER SCIENCE ; M.Sc. (COMPUTER SCIENCE)
KEYWORD : SIGNATURE FILE / MULTI-VALUED ATTRIBUTE /
RELATIONAL DATABASE

PAIRUT LEELAHAKARNJANA : USING SIGNATURE FILE TECHNIQUE FOR
SEARCHING A MULTI-VALUED ATTRIBUTE IN RELATIONAL DATABASE. THESIS
ADVISOR : DAMRAS WONGSAWANG Ph.D. 78 p. ISBN 974-661-055-4

Signature file is an indexing technique for searching unformatted data in an information retrieval system. The earliest structure of signature file is sequential signature file. In query evaluation, sequential signature file requires accessing all bits in the signature file. Many researches proposed various organizations for the signature file that try to access without all bits. One interesting organization is vertically partitioned signature file. Recently, the multi-fragmented signature file is proposed for fast searching in the multi-term query. It is also grouped in the vertically partitioned signature.

The multi-valued attribute in relational database can be viewed as unformatted data, and querying the multi-valued attribute by using SQL (Structural Query Language) requires high processing time. For these reasons, we propose a model that uses signature file for searching the multi-valued attribute in a relational database. Generally, if a relation has the multi-valued attribute, it should be split into many relations for getting normal forms. In our proposed model, the multi-valued attribute can be kept in a single relation. The signature file is created to be the index of the multi-valued attribute. We simulate the test environments and compare the results between our approach with the SQL statement approach that use SQL statement for getting the same result. We found that our approach gives faster response time than the SQL statement approach. The reason behind this result is because signature file processing deals with bit operation which always takes less processing time than record operation used in SQL.

For SQL statement approach, the factors that affect the query response time are database size and number of terms in the query. While in our approach, the query response time depends on database size, number of projects in query, and number of records in the output. Increasing database size usually increases the query response time in both approaches. Increasing the number of terms in query increases the query response time in SQL statement approach, but it decreases the query response time in our approach. The reason is because increasing the number of terms in query always reduces the number of records in output and the increased time when increasing number of terms in query is always less than the decreased time when decreasing number of records in output. Therefore, for our approach, the query response time is always decreased when the number of terms in a query is increased.