

Query optimization in databases term paper

[Business](#), [Management](#)



\n[[toc title="Table of Contents"](#)]\n

\n \t

1. [Query optimizer architecture](#) \n \t
2. [Rewriter](#) \n \t
3. [Planner](#) \n \t
4. [Reference](#) \n

\n[/toc]\n \n

Query optimizer architecture

Given a database and a query on it, there a number of processes through which the query is run through with all the processes having different execution plans. In the optimization process the best plan is selected after careful consideration of all the option plans available.

Query optimizer architectural definition gives a better understanding of the tests that are done on the alternatives give (Epstein, 2008). Below is a simple architectural view of the query optimization process.

In the simple architecture above, a classification is given based on two stages of the model; the rewriter and the planning stages. Each stage has its own module(s).

Rewriter

This stage operates in the declarative level. It is concerned with transformation of queries passed in to well-defined and efficient queries. An example of the activity done in this stage is the removal of view declaration in the query and replacing them with more efficient definition of the same.

Rewriter stage cares not for the cost of implementation of the transformed query. All assumption is made that the transformed query is more efficient than the original and hence more useful in nature.

Planner

Examination of all possible execution plans for each query is done in this stage. It is at this stage that the cost of implementation of the query is taken into consideration. Search strategies are implemented here (Sneirderman, 2006). Search strategy is important in the determination of the execution alternatives in a particular way in an execution space. Execution space consists of the algebraic space and the method structure space.

Algebraic space is a sub-module in the planner module. It is to be considered that the importance of this module is necessary. It gives the determination of the necessary order in which the planner uses in determination of query execution. Algebraic module is coupled with formulae and logic algebra that are used in the determination of the order of execution (Raghu, 2003).

The method structure space works hand in hand with the algebraic space in determination of implementation choices existing in each series of ordered choices. Based on the algebraic joint methods that are there a choice is made (Cheung, 2007). The best choice will reflect the types of data structure and their suitability of the implementation. Execution plans based on the given algebraic formulae or at least a tree formula in the algebraic space is produced by this module. A specification of each algebraic operator and the related use of search indices' are also defined.

One of the aims of query optimization is to reduce the cost of tasks done. This module ensures the realization of the same. For every stage in query optimization process, a given formulae is used in the determination of the cost and the cheapest alternative selected as the preferred plan (Davis, & Kunii, 2005). In cost management the most considered resource in the process is the buffer management, disk CPU overlap and order of operations whether sequential or random (Raghu, 2003).

The cost model needs to know the size and frequencies of database elements for the determination of cost. The query size and logical relations need also to be known in order for the same effective cost analysis determination by the cost module. Size-distribution estimator does these tasks. Statistics are also maintained by this module.

Reference

- Chen, P. s., & Akoka, J. (2006). Optimal design of distributed information systems. *IEEE trans. compute. c-29*, 12, 1068-1080.
- Chesnais, A., Gelenbe, E., and Mitrani, I. (2006). On the modeling of parallel access to shared data. *Communication. ACM* 26, 3 (Mar.), 196-202.
- Cheung, T.-Y. (2007). A method for equijoin queries in distributed relational databases. *IEEE Trans. Comput. C-31*, 8, 746-751.
- Chiu, D. M., and Ho, Y. C. (2004). A methodology for interpreting tree queries into optimal semijoin expressions. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (Santa Monica, Calif., May

14-16). ACM, New York, pp. 169-178.

Davis, L. S., & Kunii, T. L. (2005). Pattern databases. In Data Base Design Techniques

II, S. B. Yao and T. L. Kunii, Eds. Springer, New York, pp. 357-399.

Dayal, U. (2003). Evaluating queries with quantities: A horticultural approach. In

Proceedings of the ACM Symposium on Principles of Database Systems {Atlanta, Ga.).

ACM, New York, pp. 125-136.

Epstein, A., Stonebraker, M., and Wong, E. (2008). Distributed query processing in a relational data base system. In Proceedings of the ACM-SIGMOD International

Conference on Management of Data. ACM, New York, pp. 169-180.

Garcia-Molina, H., Ullhman, J., and, Widom, J., (2004) Database Systems-The Complete

Book, Pearson Education,

Raghu R., and Johannes G., (2003) Database Management Systems, 3rdEdition, McGraw Hill,

Rothnie, J. B. (2004). An approach to implementing a relational data management system. In Proceedings of the ACM-SIGMOD Workshop on Data Description, Access, and Control (Ann Arbor, Mich., May 1-3). ACM, New York, pp. 277-294.

Selinger, et. al. (2009). Access path selection in a relational database management system. In Proceedings of the ACM-SIGMOD International Conference on Management of Data (Boston, Mass., May 30-June 1). ACM,

New York, pp. 23-34.

Severance, D. G., and Carlis, J. V. (2007). A practical approach to selecting record access paths. *ACM Comput. Surv.* 9, 4 (Dec.) 259-272.

Shneiderman, B., and Goodman, V. (2006) Batched searching of sequential and tree structured files. *ACM Trans. Database Syst.* 1, 3 (Sept.), 268-275.

Walker, A. (2007). On retrieval from a small version of a large data base. In *Proceedings of the 6th International Conference on Very Large Data Bases* (Montreal, Oct. 1-3). IEEE, New York, pp. 47-54.

Warren, D. H. D. (2005). Efficient processing of interactive relational database queries expressed in logic. In *Proceedings of the 9th International Conference on Very Large Data Bases* (Cannes, Sept. 9-11). IEEE, New York, pp. 272-281