SQL-query Syntactic Optimization Based on the Local Model of the Controlled Process

Pavlo Kostenko, Kremenchuk Mykhailo Ostrogradskyi National University
(01.11.2009, ScD, prof. Guchenko Mykola, Kremenchuk Mykhailo Ostrogradskyi National University)

Abstract
This paper describes the data storage technology (relation databases) and the query language to these databases (Structured Query Language), which today are the basis of the all information systems evolution. The article contains the review of the existing techniques for SQL-query optimization. Mathematical expressions for the generation of possible SQL-query syntactic constructs are considered. The paper contains the main control characteristics of the dynamic objects under conditions of low apriori information, based on the local model of the controlled process. The new approach to automatic external syntactic optimization of the SQL-query is presented.

1. Introduction
Database management systems (DBMS) have become an integral part of modern computer information systems (CIS) (Fig.1). Actual and comprehensively common part of CIS is an automated control system workplace (ACSW). Such systems are used for accelerating the organization and operation of complex, multi-iterative and branching production processes, particularly in medical, legal institutions, accounting, economic, statistical departments.

2. Ways to improve quick-action of the CIS
Following factors affects on the information rate in CIS: software implementation of the front-end CIS; hardware specifications of the front-end and back-end computers and data network; database structure; database query; etc. Acceleration of the CIS quick-action is seen in optimization of the mentioned factors.

Refactoring methods are used more often than optimization methods for the front-end of the CIS [1,2], due to their complexity and source codes saturation. Software source codes optimization leads to deterioration of their logical and syntactic understanding, so optimization at this level is performed more rarely.

Hardware optimization is reduced to systematic replacement (upgrade) of hardware parts. Hardware characteristics, interaction methods and increasing of the computing possibilities of it elements are considered in [3,4].

Database structure optimization requires the information reorganization in a database on a logical level (database normalization) [5].

Normalization means reduction information in the database to the normal forms. It deallocates database from redundant data and ensures data integrity. Currently there are six normal forms [6].

Normalization forms can reduce data redundancy, the appearance of their duplicates, involve support methods of the data integrity. As a result, the conflicting data probability is significantly reduced, administration and information updating are simplified, the amount of disk space is reduced.

But such optimization leads to the construction of the "very" complex queries. Data accessing is too slower down, because query has a large number of the tables’ connections. To speed up the data selection and simplify the programming queries often it is necessary to carry out a selective database denormalization and optimization of the SQL – queries.
General methods and algorithms for optimizing SQL - queries are considered in [7-13]. Speaking about queries optimization in the relational databases, usually we mean a way of queries processing when on the basis of initial query through transformation a procedural plan of the query implementation is produced. This plan is optimal for the inside database control structures. Corresponding transformation of the initial query is done by a DBMS special component - optimizer. Optimality of the produced query plan is conditional: the plan is optimal according to the criteria laid down in the optimizer, which generates a possible difference from the real optimum.

Query optimizer chooses the "best" way to execute the query on the basis of the known strategies of the elementary query constituents and composition of the more complex strategies based on elementary. Thus, the search space of the optimal query execution plan is limited by the pre-fixed basic strategies.

Inside SQL Optimizer has all the necessary data for correct Query Evaluation Plan (QEP) optimization (metadata database, indexes, statistics etc.), the only weak point of this system is the optimization rules. Default optimizer is limited in his work with just one query, which results in limited number of the optimization options. The choice of not optimal QEP is conditioned by incorrect form of the user query. Query modification on the client-side is implemented by technology of refactoring that described in [14]. Refactoring of the SQL applications and queries is a laborious process, which requires a periodic application, due to the dynamic complication of the database structure and its volume. Therefore we have the problem of the optimization structure of the SQL - queries in real time, before it's done on the database server, during CIS work.

3. Strategies for optimization SQL - queries

Dr. E.F. Codd fundamental work [15] of the relational data model theory and his followers, including M. Meyer [16], is one of reasons of widespread use of the relational databases. Relational data model combines the mathematical and logical models. These models establish the rules of the organization and work with information stored in the database.

Advantages of the relational data model theory include: strict logical formalization, data independence and the possibility of the optimization structure of the database using normalization. Getting information from the database in most cases is implemented with SQL - queries.

Declarative programming language SQL - is not a procedural language that significantly affects on its work efficiency.

The logical query optimization, query parsing and selecting the optimal execution (evaluation) plan have the largest share in the usage of computational resources and in influence on the processing time. Development of the optimal QEP generation system is the actual task from the early 1980's.

Four strategies for the optimization queries:
- Bottom-Up Optimizers.
- Top-Down Optimizers.
- Hybrid Optimizers (Bottom-Up/Top-Down).
- Genetic Optimizers.

3.1 Bottom-Up Optimizers

Historically, the Bottom-Up strategy is considered as the first search strategy, it's based on the dynamic programming method. The System R optimizer is a representation of this strategy [17]. Method of the dynamic programming finds the optimal plan by $O(n^n)$ sorting, which considerably faster than direct busting $O(n!)$ plans.

Lohman's G. work [18] about creating rules usage for generation of the plans in bottom-up optimizer system was implemented in Starburst [19]. The structural representation of the SQL - query, that is Query Graph Model (QGM), was firstly applied in Starburst. Starburst uses a set of the heuristic rules to convert QGM - query into semantically appropriate and "best" model QGM, because of the excessiveness minimization and obtaining of the query form that is better optimized to the stage of the plan generation. Revolutionary characteristics of this optimizer are additionally confirmed by the introduction in the structure generator the parametric transformation rules for building implementation plans. They are called "Strategy Alternative Rules» (STARs), and determines the different aspects of the performance: the internal table representation, combining algorithms, etc. Query optimizer, on the basis of a given set of rules, generates a set of QEP, which are formed with a set of the embedded primitive calls, predetermined operators (LOLEPOP - Low-Level-Plan-Operator).

Disadvantages:
1. Dynamic programming algorithm has exponential dependence of the required memory volume from the number of combined tables. This fact limits the queries optimization with a combination of more than 15 tables.
2. Heuristic rules are based only on logical information but not on the cost evaluation, it sometimes can lead to the selection of the nonoptimal QEP [20].
3. Optimization of the complex non-relational operators requires the significant laborious adaptation of the Starburst.

3.2 Top-Down Optimizers

Articles of the optimizers Exodus, Volcano, Cascades. Graefe G. had proposed firstly to use the environment of the top-down optimizing queries based on the separation principle of the search strategy and data model. The principles in the optimizer Exodus had accelerated the development of the scalable optimizers of the next generations.

In the Volcano optimizer was the idea to increase the DBMS productivity by using algorithm "memorization". The best plan search, on the basis of the top-down controlled strategy, optimizes sub-queries only by their affiliation to optimal sub-plan.  

Disadvantages: Built search strategy had led to the equivalence of the Volcano algorithm productivity to the Starburst due to the first generation of a set of possible logic expressions and second generation of the physical expressions.

3.3 Hybrid Optimizers

The development of the object-oriented programming, plan costs method, index fields, etc. had led to the appearance of new methods for queries optimization. Prominent examples of such optimizers are Cascades [22] and OPT ++ [23].

In addition, OPT ++ system includes the principles of the Bottom-Up and Top-Down optimization and it is based on the comparative analysis of their performance. Hybrid search strategies have following properties [21]:

- optimization tasks are represented as data structures;
- rules are represented as objects;
- availability of the additional rules;
- priority sorting of the rules;
- predicates representation by the elements of the search tree;
- usage of the interrupt rules for operators, cost models and optimization rules.

The apparent advantage is the usage of "memorization" technique for group optimization. Unlike the algorithm Volcano, which always generates all equivalent logical expressions in the preparatory stage before the optimization stage, Cascades tests the group only if the absence of the optimized plan version is confirmed.

Disadvantages: Usage of virtual functions, methods of the dynamic allocation and deallocation of the memory, a significant separation of a medium optimizer and database administration software, extremely low probability of the turning on the "memorization" groups’ algorithm.

3.4 Genetic Optimizers

Partially mentioned disadvantages are solved by optimization systems of the fourth generation [24-26]. Modern optimizers are based on the support of the reliable statistical information about columns values distribution in the tables. Those tables are stored in databases and dynamically generated at queries execution. Statistical information is used in the subsequent formation of the optimal plans.

In the mentioned area the IBM approach is considered as fundamental, it is represented by experimental adaptive optimizer LEO [27]. Optimizer calculates the corrections for the estimations and conducts the comparative analysis of the estimations with the actual capacity of the result at each step of the QEP during the constant monitoring of the query execution. These corrections can be used to optimize similar queries in the future. LEO initiates re-optimizing of the query in the middle of its execution if the estimations’ errors are detected. LEO collects data about power of the results obtained by accessing to the tables, and corrects the estimations’ errors for simple predicates by harmonizing the databases statistics with the expectation of the future queries.

Alternative direction of the optimization systems development is usage of the genetic algorithms. RDBMS PostgreSQL actively uses principles of the genetic optimization [28]. Optimizer GEQO is partly based on genetic D. Whitley’s algorithm. Technological feature of this approach is the solution of the queries optimization problem with modernized algorithm of the travelling salesman. Possible plans for query execution are represented in a tree with weighting coefficients. Rapid improvement of the query plan is achieved with using the stationary states of the genetic algorithm (replacing are conducted only in individual links of the tree, with creation a new mutation).

Disadvantages:

1. Disadvantage of the adaptive (genetic) optimizers is two-level mechanism commissioning. The first level requires a long learning with significant probability of the nonoptimal QEP. The second level assumes a gradual disconnection of the adaptive mechanism and selection of the optimal queries based on reliable statistical correlation and facts.

2. Constant mode without long enough statistical data collection leads to QEP selection only by reliable facts, despite the optimal QEP.

3. Optimal plan selection for a single query is the nondetermined (aperiodic) process because of a random initial plan selection and slow "mutations" in the optimization process. Consequently, completely different plans with significantly different execution times can be used for a single query.
Analysis of the SQL - query optimizing algorithms has shown their significant limitations and many disadvantages. In particular, the limited expansion at the strict organization, rejection of the query execution plan due to a limited heuristics sets, limitations in queries optimization that include a combination of more than 15 tables, RAM overloading with numerous appeals regarding dynamic memory allocation and release during the optimization, complexity and long setting into operation etc. All considered optimization systems are dependent on the structure and database (metadata) characteristics, are an integral part of the DBMS and are limited in the context of the structural modernization by DBMS administrator. Therefore, it is actual to develop an optimizer, which will be independent of the DBMS and the database structure (metadata), will require a minimum study period, and will consider the queries optimization as a periodic process that will eliminate the disadvantages of genetic optimizers.

4. SQL - optimizer based on local model of the controlled process

A possible solution to this problem is to use adaptive control system. But small amount of a priori information, a significant current information uncertainty and the need to control in real-time mode lead to the difficulties in usage of the statistical, intelligent and robust control systems. Experiment in real time mode makes it possible to obtain a model of controlled process [29]. This local model presents the dynamics of control object and disturbances acting on it in current moment.

Optimizer, developed on this principle, is not able to use data access methods such as: relations scanning, index search (B-tree), index scan (scan B-tree leaves), access by unique identifiers of the physical layer, search for index cluster, search in the hash - cluster, search for the bit index, search for the join bit index.

Optimization process is based on syntactic and semantic optimization, with heuristic rules. Heuristic rules are widely used in existing systems. But if a single "optimized" plan is based on them, it usually leads to the selection of the inefficient query execution plan.

After users’ query execution a conscious deterioration of the conditions of the data selection is made for better understanding of the structure and database loading (Fig. 2). Execution time of the deteriorated query and user query serve as a criterion for queries optimization and allow to exclude the incorrect versions from further optimization cycles.

During optimization process the generation of the unique syntax query construction is performed when its semantic construction is respectively equal to initial query. Algorithm selects the major syntactic constructions by sorting terms. If sorting terms are missing the number of the major structures is equal to the number of conditions in the block «WHERE» of the SQL-query.

To calculate number of the possible syntactic structures we introduce the following notations: k - number of tables, n – connections number in the FROM area, m - conditions number in the area WHERE, x - nodes number in the query graph, X - the conditions number for the query graph nodes, y - branches number in the query graph, z - number of the sorting fields, Ni - conditions number for i-th connection in the zone FROM.

Total number of syntax constructions for a simple query without conditions in block "join" is equal to (1). With conditions in the block "join" - (2). If the conditions from block "join" are moved to block "where" – (3), (without using explicit connections).

\[
\prod_{i=1}^{n} N_{i-1} \times m \times n!
\]

\[
(\frac{m + n + \sum_{i=1}^{n} N_{i-1}}{n})!
\]

If the explicit connections and selected data conditions are moved from block "join" to block "where":

\[
\sum_{i=1}^{n} \left( \left( \sum_{j=1}^{n} N_{j-1} \right) \times (n-i)! \times \left( m + i + \sum_{w=1}^{i} N_{w-1} \right) \right)
\]
Certain syntactic query constructions require the constant sequence of tables’ connections. In this case, the total number of the syntactic structures is calculated by the formula (5).

It is experimentally found in [30] that the sequence of tables’ connections does not significantly affect on the query execution speed. If the explicit connections and selected data conditions are completely moved from block “join” to block “where” than formed implicit connections are considered as one condition. If the sorting is taken into account we have formula (6). The number of necessary iterations for optimal query search is in (7):

$$
C_{sql} = \begin{cases}
  m! & \text{if } z = 0, \forall N > 1 \\
  m! + \left( \sum_{i=1}^{n} N_{i-1} \right)! & \text{if } z > 1, \forall N > 1
\end{cases}
$$

(7)

For the average query with parameters: \( k = 5 \), \( n = 4 \), \( m = 3 \), \( x = 2 \), \( X = 1 \), \( y = 3 \), \( z = 1 \), \( N_{i} = \{0,0,0,0\} \), taking into account the properties of the proposed algorithm, search of the optimal query requires the generation and inspection of the six queries’ structures.

5. Result, conclusions, future work

The analysis of the existing strategies for SQL-queries optimization is carried out. Their advantages and disadvantages are pointed out. Alternative method of the SQL-queries optimization based on local model of the controlled process is presented. Proposed method allows to avoid the limitations of the structural optimization in syntax constructions. Formulas for calculation of the total number of the syntactic structures at query optimization are formed. On the basis of his previous works, author proposes the formula for calculation the minimum number of the syntactic structures to optimize a query. Experiments on the DBMS FireBird 2.0, which is not presented in this paper, confirm the effectiveness of the proposed algorithm. Further work is seen in the development of the automated interception software and queries optimization in real time.

Bibliography


Authors:
MSc. Pavlo Kostenko
Kremenchuk Mykhailo
Ostrogradsky National University
ul. Pershotravneva 20
39600 Kremenchuk
tel. +38(050) 958 15 39
fax. 38 (5366) 3 60 00
e-mail: ppkostenko@gmail.com