INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

SQL query database manager tool for developer

Abhishiktha. K
kabhishiktha15@gmail.com
SRM Easwari Engineering College, Chennai, Tamil Nadu

Krithika. S
kikisub1997@hotmail.com
SRM Easwari Engineering College, Chennai, Tamil Nadu

Pradeep. R. S
pradeep.rs@hotmail.com
SRM Easwari Engineering College, Chennai, Tamil Nadu

Dr. G. S. Anandha Mala
kabhishiktha15@gmail.com
SRM Easwari Engineering College, Chennai, Tamil Nadu

ABSTRACT

To design the database and tune SQL Queries. Tuning can be done by reducing the total CPU time and reducing the input taken by the Query. This tool can be used by the developer to tune the queries and to examine Nested and normalized queries, and information about database objects. The input to the optimizer is a parsed SQL query and statistics about the tables, indexes, and columns named in the query. The optimizer examines parsed and normalized queries, and information about database objects. The input to the optimizer is a parsed SQL query and statistics about the tables, indexes, and columns named in the query. The output from the optimizer is a query plan. The query plan is compiled code that contains the ordered steps to carry out the query, including the access methods (table scan or index scan, type of join to use, join order, and so on) to access each table. Using statistics on tables and indexes, the optimizer predicts the cost of using alternative access methods to resolve a particular query. It finds the best query plan – the plan that is least costly in terms of input.

Keywords: *SQL Query, Database Manager Tool, SQL tuner, SQL (Structured Query Language), SQL Optimization.*

1. INTRODUCTION

This project is developed for designing and maintaining SQL database. And also tuning can be done by reducing the total CPU time and also reducing the I/O taken by the Query. This tool can be used by the developer to create database and tune the queries. There is no tool that helps the developer to create and maintain database without the help of DBA (Database Administrator) This SQL query manager tool makes the easiest way to design and maintain the database and it helps in tuning the SQL queries by Cost based method and Execution plan methodology. The optimizer examines parsed and normalized queries, and information about database objects. The input to the optimizer is a parsed SQL query and statistics about the tables, indexes, and columns named in the query. The input to the optimizer is a parsed SQL query and statistics about the tables, indexes, and columns named in the query. The output from the optimizer is a query plan. The query plan is compiled code that contains the ordered steps to carry out the query, including the access methods (table scan or index scan, type of join to use, join order, and so on) to access each table.

2. RELATED WORKS

"Crowd Op: Query Optimization for Declarative Crowd Sourcing Systems", by the authors Meihui Zhang, Kok.S, Meiyu Lu, We study the query optimization problem in declarative crowdsourcing systems. Declarative crowdsourcing is designed to hide the complexities and relieve the user the burden of dealing with the crowd. The user is only required to submit an SQL-like query and the system takes the responsibility of compiling the query, generating the execution plan and evaluating in the crowdsourcing marketplace. A given query can have many alternative execution plans and the difference in crowdsourcing cost between the best and the worst plans may be several orders of magnitude. Therefore, as in relational database systems, query optimization is important to crowdsourcing systems that provide declarative query interfaces. In this paper, we propose

CROWDOP, a cost-based query optimization approach for declarative crowdsourcing systems. CROWDOP considers both cost and latency in the query optimization objectives and generates query plans that provide a good balance between the cost and latency. We develop efficient algorithms in the CROWDOP for optimizing three types of queries: selection queries, join queries and complex

selection-join queries. We validate our approach via extensive experiments by simulation as well as with the real crowd on Amazon Mechanical Turk.

"Robust Query Processing through Progressive Optimization",Volker Markl, Vijayshankar Raman, David Simmen, Guy Lohman, Hamid Pirahesh, Miso Cilimdzic introduce Virtually every commercial query optimizer chooses the best plan for a query using a cost model that relies heavily on accurate cardinality estimation. Cardinality estimation errors can occur due to the use of inaccurate statistics, invalid assumptions about attribute independence, parameter markers, and so on. Cardinality estimation errors may cause the optimizer to choose a sub-optimal plan. We present an approach to query processing that is extremely robust because it is able to detect and recover from cardinality estimation errors. We call this approach "progressive query optimization" (POP). POP validates cardinality estimates against actual values as measured during query execution. If there is significant disagreement between estimated and actual values, execution might be stopped and re-optimization might occur. Oscillation between optimization and execution steps can occur any number of times. A re-optimization step can exploit both the actual cardinality and partial results, computed during a previous execution step. Checkpoint operators (CHECK) validate the optimizer's cardinality estimates against actual cardinalities. Each CHECK has a condition that indicates the cardinality bounds within which a plan is valid. We compute this validity range through a novel sensitivity analysis of query plan operators. If the CHECK condition is violated, CHECK triggers re-optimization. POP has been prototyped in a leading commercial DBMS. An experimental evaluation of POP using TPC-H queries illustrates the robustness POP adds to query processing, while incurring only negligible overhead. A case-study applying POP to a real-world database and workload shows the potential of POP, accelerating complex OLAP queries by almost two orders of magnitude.

"Adaptive Query Processing in the LookingGlass",Babu,Shivnath and Bizarro,Pedro, A great deal of work on adaptive query processing has been done over the last few years: Adaptive query processing has been used to detect and correct optimizer errors due to incorrect statistics or simplified cost metrics; it has been applied to long-running continuous queries over data streams whose characteristics vary over time; and routing-based adaptive query processing does away with the optimizer altogether. Despite this large body of interrelated work, no unifying comparison of adaptive query processing techniques or systems has been attempted; we tackle this problem. We identify three families of systems (*plan-based*, *CQ-based*, and *routing based*), and compare them in detail with respect to the most important aspects of adaptive query processing: plan quality, statistics monitoring and re-optimization, plan migration, and scalability. We also suggest two new approaches to adaptive query processing that address some of the shortcomings revealed by our in-depth analysis: (1) *Proactive-optimization*, where the optimizer chooses query plan22s with the expectation of re-optimization; and (2) *Plan logging*, where optimizer decisions under different conditions are logged over time, enabling plan reuse as well as analysis of relevant statistics and benefits of adaptivity.

"External optimization of SQL-query under conditions of databases structural uncertainty", the authors M. Zagirnyak, P. Kostenko, paper deals with factors that influence the speed of data receiving in information systems. The method of automatic external SQL-query optimization is described. The method is based on the principle of building a local model of controlled process. It allows one to optimize the SQL-queries regardless of applied database management system and its settings. The structural and functional diagram of adaptive system of an external SQL-query optimization is presented. Possibility of using the proposed method under the conditions of databases structural uncertainty was tested. Electric network architecture.

"Query optimization over crowdsourced data", Hyunjung Park, Jennifer Widom, propose Deco is a comprehensive system for answering declarative queries posed over stored relational data together with data obtained on demand from the crowd. In this paper we describe Deco's cost based query optimizer, building on Deco's data model, query language, and query execution engine presented earlier. Deco's objective in query optimization is to find the best query plan to answer a query, in terms of estimated monetary cost. Deco's query semantics and plan execution strategies require several fundamental changes to traditional query optimization. Novel techniques incorporated into Deco's query optimizer include a cost model distinguishing between "free" existing data versus paid new data, a cardinality estimation algorithm coping with changes to the database state during query execution, and a plan enumeration algorithm maximizing reuse of common sub-plans in a setting that makes reuse challenging. We experimentally evaluate Deco's query optimizer, focusing on the accuracy of cost estimation and the efficiency of plan enumeration.

"Process Optimization with Consideration of Uncertainties-An Overview", YingChen, ZhihongYuan, Bingzhen Chen, Optimization under uncertainty is a challenging topic of practical importance in the Process Systems Engineering. Since the solution of an optimization problem generally exhibits high sensitivity to the parameter variations, the deterministic model which neglects the parametric uncertainties is not suitable for practical applications. This paper provides an overview of the key contributions and recent advances in the field of process optimization under uncertainty over the past ten years and discusses their advantages and limitations thoroughly. The discussion is focused on three specific research areas, namely robust optimization, stochastic programming and chance constrained programming, based on which a systematic analysis of their applications, developments and future directions are presented. It shows that the more recent trend has been to integrate different optimization methods to leverage their respective superiority and compensate for their drawbacks. Moreover, data-driven optimization, which combines mathematical programming methods and machine learning algorithms, has become an emerging and competitive tool to handle optimization problems in the presence of uncertainty based on massive historical data.

"Data allocation optimization for query processing in graph databases using Lucene", Anita Brigit Mathew, proposed Methodological handling of queries is a crucial requirement in social networks connected to a graph No SQL database that incorporates massive amounts of data. The massive data need to be partitioned across numerous nodes so that the queries when executed can be retrieved from a parallel structure. A novel storage mechanism for effective query processing must to be established in graph databases for minimizing time overhead. This paper proposes a meta-heuristic algorithm for partitioning of graph database

across nodes by placement of all related information on same or adjacent nodes. The graph database allocation problem is proved to be NP-Hard. A meta-heuristic algorithm comprising of Best Fit Decreasing with Ant Colony Optimization is proposed for data allocation in a distributed architecture of graph No SQL databases. Lucene index is applied on proposed allocation for faster query processing. The proposed algorithm with Lucene is evaluated based on simulation results obtained from different heuristics available in literature.

"A JIT Compilation-based Unified SQL Query Optimization System" Myungcheol Lee, Miyoung Lee, ChangSoo Kim proposed, In-memory databases are gaining attention as a solution to efficiently support SQL queries on large volume of data, as main memories are becoming cheaper and grow in size. However, their query performance is not well improved on modern hardware with faster CPUs, registers and caches due to the limitation of the classical iterator style query processing model. We propose a unified SQL query optimization system using JIT compilation of OLTP, OLAP, and Stored Procedure workloads for enhanced performance on modern hardware.

"Scalable Query Optimization for Efficient Data Processing Using MapReduce", Yi Shan, Yi Chen, MapReduce is widely acknowledged by both industry and academia as an effective programming model for query processing on big data. It is crucial to design an optimizer which finds the most efficient way to execute an SQL query using MapReduce. However, existing work in parallel query processing either falls short of optimizing an SQL query using MapReduce or the time complexity of the optimizer it uses is exponential. Also, industry solutions such as HIVE, and YSmart do not optimize the join sequence of an SQL query and cannot guarantee an optimal execution plan. In this paper, we propose a scalable optimizer for SQL queries using MapReduce, named SOSQL. Experiments performed on Google Cloud Platform confirmed the scalability and efficiency of SOSQL over existing work.

"Efficient Processing and Optimization of Queries with Set Predicates using Filtered Bitmap Index",Jayant Rajurkar,T.K.Khan, proposed In today's complex world requires state-of-the-art data analysis over massive data sets. In data warehousing and OLAP applications, scalar-level predicates of set in SQL become highly inadequate which needs to support set-level comparison semantics, i.e., comparing a group of tuples with set of values. Complex queries composed by scalar-level operations are challenging for database engine to optimize, which results in costly evaluation. Bitmap indexing provides an important database capability to accelerate queries. Few database systems have implemented these indexes because of the difficulties of modifying fundamental assumptions in the low-level design of a database system. Bitmap index built one bitmap vector for each attribute value is gaining popularity in both column-oriented and row-oriented databases. It requires less space than the raw data provides opportunities for more efficient query processing. In this paper, we studied the property of bitmap index and developed a very effective bitmap pruning strategy for processing queries. Such index-pruning-based approach eliminates the need of scanning and processing the entire data set and thus speeds up the query processing significantly. Our approach is much more efficient than existing algorithms commonly used in row-oriented and column oriented database.

3. SYSTEM TUNING

A. Performance Tuning

Performance tuning is an important part of today's database applications. Very often, large savings in both time and money can be achieved with proper performance tuning. The beauty of performance tuning is that, in many cases, a small change to an index or a SQL query can result in a far more efficient application.

B. Query Optimization

Query optimization is an important skill for SQL developers and database administrators (DBAs). In order to improve the performance of SQL queries, developers and DBAs need to understand the query optimizer and the techniques it uses to select an access path and prepare a query execution plan. Query tuning involves knowledge of techniques such as cost-based and heuristic-based optimizers, plus the tools an SQL platform provides for explaining a query execution plan.

C. SQL Tuner

Main Objective of **SQL Tuner** is to **analyze** the query provided by the user and **suggest** them the ways by which they can **optimize** the query for performance.

SQL Tuner is a tool principally made for DBA's and to minimize their work load, developers can also use them to prepare the optimized.

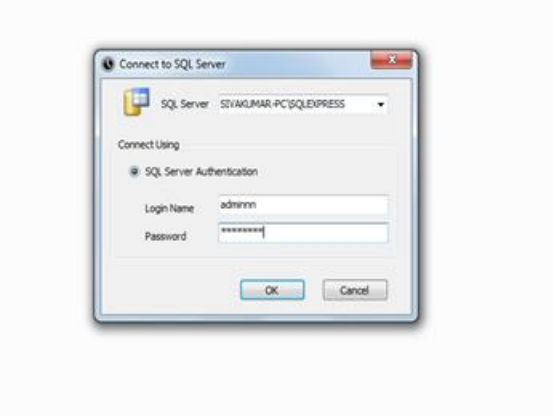
4. SYSTEM IMPLEMENTATION

The normal scenario in today's world is that the user or the programmer who is using the query for working with the database often faces the problems of slow execution of the query. One reason for this problem could be that many users are trying to access the SQL Server at the same time. But user can reduce the time taken by the SQL Server for the execution of the query by optimizing the query.

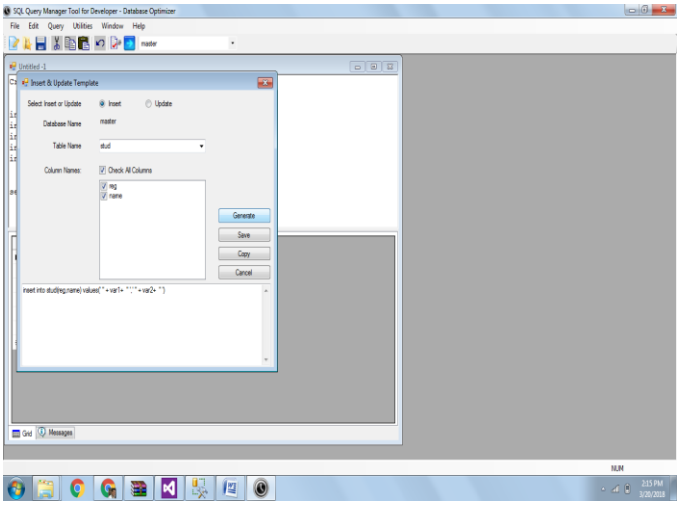
Now the problem in optimizing is that user or programmer is not well versed with SQL Server Optimizer and the way in which it executes its query. So what the user or the programmer does is that it submits it query to the DBA of the company or of an organization. The DBA has to go through the query and change the query or the table in such a way that the query performs in an optimum way. Even DBA were not able to tune to the query of the user very efficiently and even if it does the tuning efficiently it would take lots of time and resources of the DBA's. This wouldn't be feasible at the time when queries are urgently needed by the user or the programmer.

So one way was to get a tool in which the query would get optimized and that also in less time and also without the help of the DBA. Even one more problem that the DBA was faced was that he has to rely on his own skills and experience to optimize the query as there are no set rules for doing the optimization. This would also cause a problem as there in no reference against which the DBA can be assured that the query is been optimized properly.

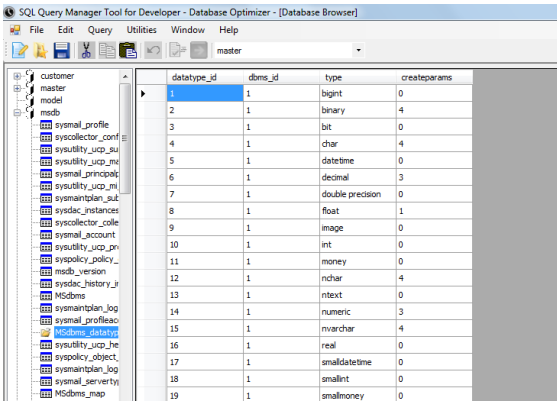
A. SQL Server Connectivity



B. Query Generator



C. Data Browser



5. CONCLUSION AND FUTURE ENHANCEMENT

We have accomplished in generating the necessary SQL code needed for a particular scenario by developer for his front end. We use various basic SQL commands in generating the necessary code. A grid view for easy access and updation of data is done effectively. This in effect with code optimization is also done.This project was developed to understand how the SQL Server Optimizer optimizes the queries and reduces the query's CPU time and Input\Output required.

So there are many things for future enhancements of this project. The future enhancements that are possible in the project are as follows:

- To optimize more complex queries i.e. queries which include Joins, Unions, Sub Queries etc.
- To study the database structure and provide the user with suggestions to improve the database structure for best performance.
- To optimize the query which is been embedded in the Application, without the efforts of the user or programmer entering the query in the SQL Tuner.

6. REFERENCES

[1] Meihui Zhang, Kok.S, Meiyu Lu, Crowd Op: Query Optimization for Declarative Crowd sourcing Systems, IEEE Transactions on Knowledge and Data Engineering, Volume 27 Issue 8 March 2015
[2] V. Markl, V. Raman, D. Simmen, G. Lohman, H. Pirahesh, Robust Query Processing through Progressive Optimization, In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, 2004, pp. 659–670
[3] Babu, Shivnath and Bizarro, Pedro (2005) Adaptive Query Processing in the Looking Glass. In: Second Biennial Conference on Innovative Data Systems Research (CIDR 2005), January 4-7, 2005, Asilomar, California.
[4] M. Zagirnyak, P. Kostenko External optimization of SQL-query under conditions of databases structural uncertainty, IEEE Journal, Volume - 17, Issue -23, Pages: 7828 – 7837, 2017.
[5]H. Park and J. Widom. Query optimization over crowdsourced data.PVLDB, 6(10):781–792, 2013, IEEE Trans. Magn. 48 (2012) 259–262.
[6] YingChen,ZhihongYuan,Bingzhen Chen, Process Optimization with Consideration of Uncertainties-An Overview IEEE Journal, Volume – 4, No - 5, pp. 1563-1570, Oct. 2017.
[7] Anita Brigit Mathew Data allocation optimization for query processing in graph databases using Lucene ,IEEE Sensors Journal, Volume - 16, No - . 5, pp. 1361-1367, March1, 2016.
[8] Myungcheol Lee; Miyoung Lee; ChangSoo Kim, A JIT Compilation-based Unified SQL Query Optimization System, 2016 IEEE 6th International Conference on IT Convergence and Security (ICITCS).
[9] Yi Shan, Yi Chen, Scalable Query Optimization for Efficient Data Processing Using MapReduce, DOI: 10.1109/BigDataCongress.2015.100 Conference: Conference: 2015 IEEE
[10]Jayant Rajurkar, T.K.Khan, Efficient Processing and Optimization of Queries with Set Predicates using Filtered Bitmap Index,IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)2015

