Performance Analysis of Query Optimization In Relational Database Management System

Rima A. Patel
[1, 2] Dept of Computer Engineering
[1, 2] Vadodara Institute of Engineering, Kotambi
Vadodara, India

Abstract- *Query optimization is of great use now a days, as each and every field uses the database system to store the information. It is necessary to store the information in ample way. Performance of relational database is rely on query optimaization,which can be used in operational as well as transactional database. This paper represents the performance analysis of query optimization in relational database management system. The main focus is on query optimization techniques and the components of query optimizer.*

I. INTRODUCTION

Now days we need to deal with the hundreds of thousands of raw facts, data and information. To store the data and information, the best database model with good database structure and design is the primary requirement. The performance of data model depends on the effective and efficient retrieval of data. The performance measure of relational data model depends on the best query evaluation plan describes data to be retrieved from a database.[1] In the context of query optimization, it is often assumed that queries are expressed in a content-based (and mostly set-oriented) manner, giving the optimizer sufficient choices among alternative evaluation procedures. [1]

II. RELATIONAL DATABASE SYSTEM

A. Definition

A Relational Database Management System (RDBMS) is a software system that provides access to a relational database. The software system is a collection of software applications that can be used to create, maintain, manage and use the database. A "relational database" is a database structured on the "relational" model. Data are stored and presented in a tabular format, organized in rows and columns with one record per row..

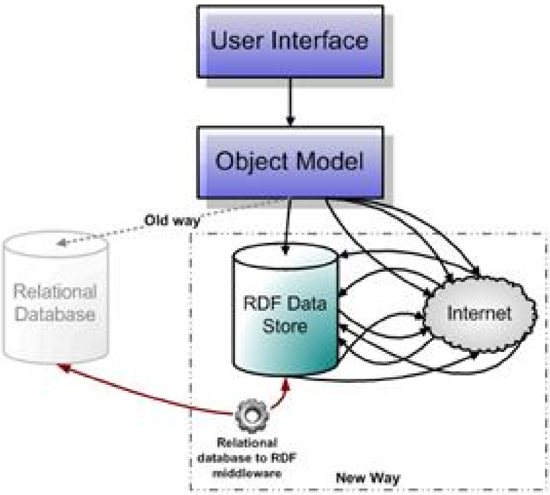
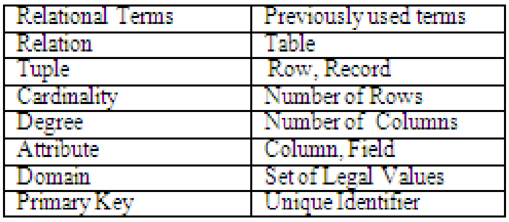


Figure.1 Relational Model.

The relational model represents the database as a collection of relations [2]. Each relation corresponds to a table of values or to some extent a flat file of records. The terms that were previously used to describe the database properties ad functions lacked the precision necessary for the formal theory that Codd was proposing [3]. So a new set of terminology had to be evolved. The following table gives the list of relational terms.

Table 1 Relational Terms



Relational Terms	Previously used terms
Relation	Table
Tuple	Row, Record
Cardinality	Number of Rows
Degree	Number of Columns
Attribute	Column, Field
Domain	Set of Legal Values
Primary Key	Unique Identifier

B.Advantages of Relational Database Management System

1. Data is only stored once

No multiple record changes needed
More efficient storage
Simple to delete or modify details.

2. Complex queries can be carried out.

A language called SQL has been developed to allow programmers to 'Insert', 'Update', 'Delete', 'Create', 'Drop' table records. These actions are further refined by a 'Where' clause. For example

SELECT * FROM Employee WHERE ID = 2

This SQL statement will extract record number 2 from the Employee table. Far more complicated queries can be written that can extract data from many tables at once.

3. Better security.

By splitting data into tables, certain tables can be made confidential. When a person logs on with their username and password, the system can then limit access only to those tables whose records they are authorized to view. For example, a student would be able to view time table and subject details but not the salary of staff.

4. Cater for future requirements.

By having data held in separate tables, it is simple to add records that are not yet needed but may be in the future. For example, the city table could be expanded to include every city and town in the country, even though no other records are using them all as yet. A flat file database cannot do this.

5. Ease of use:

The revision of any information as tables consisting of rows and columns is much easier to understand.

6. Flexibility:

Different tables from which information has to be linked and extracted can be easily manipulated by operators such as project and join to give information in the form in which it is desired.

7. Precision:

The usage of relational algebra and relational calculus in the manipulation of the relations between the tables ensures that there is no ambiguity, which may otherwise arise in establishing the linkages in a complicated network type database.

B. Disadvantages of Relational Database Management System

1. Performance: A major constraint and therefore disadvantage in the use of relational database system is machine performance.

2. Physical Storage Consumption: With an interactive system, for example an operation like join would depend upon the physical storage also. It is, therefore common in relational databases to tune the databases and in such a case the physical data layout would be chosen so as to give good performance in the most frequently run operations. It therefore would naturally result in the fact that the lays frequently run operations would tend to become even more shared.

3. Slow extraction of meaning from data: if the data is naturally organized in a hierarchical manner and stored as such, the hierarchical approach may give quick meaning for that data.

4. Data Complexity: Data in an RDBMS resides in multiple tables, which are linked to each other through shared key values. An RDBMS does not force database designers to impose a coherent table structure; inexperienced programmers may design systems that create unnecessary complexity or limit the future development of the database through poorly chosen data types. The flexibility of an RDBMS presents a double-edged sword. Experienced designers work magic, but inexperienced designers wreak havoc on a company's data.

5. Broken Keys and Records: Relational databases require shared keys to link information spread across several tables. For example, a customer table may include client demographics, with a unique index number identifying the record within the table. A sales table may identify the customer only by that index number. If the data types linking the keys are different, the database cannot link the records without additional rework by the report developer. Likewise, if a table lacks a unique key, the database may return inaccurate results. If the application accessing a database isn't coded to lock records during an edit, users could inadvertently corrupt data, leading to broken records.

6. Developer Expertise: As the complexity of a relational database increases, the skill set required by the RDBMS administrator, various users and report developers also increases. A mission-critical database may require expertise that exceeds the budget of a small business; furthermore, if the developers did not uniformly engage in best practice design, a subsequent developer may not understand hidden intricacies that could lead to broken queries or inaccurate reports. This risk increases if database and application development is performed by different people.

7. Hardware Performance: Complex queries require sophisticated processing power. Although most desktop computers can manage the databases of the size and complexity often encountered in a small business setting, a database with external data sources or very complex data structures may require more powerful servers to return results within an acceptable response time.

III. QUERY OPTIMIZATION

Query optimization refers to the process of execution of a query efficiently. This requires how to fire a given query such that it takes smaller number of operations and the low memory space. It is the most important part of the query evaluation process. Query optimization is a function in which multiple query plans for satisfying a query are examined and a good query plan is identified. There is a trade-off between the amount of time spent figuring out the best plan and the amount running the plan. The resources which are considered for costing are CPU

Path length, amount of disk buffer space, disk storage service time, and interconnect usage between units of parallelism [6]. The set of query plans examined is formed by examining possible access paths and various relational tables join techniques. These plans are generated by the parser while parsing the query. The search space can become quite large depending on the complexity of the SQL query [4,5].

A. Queries

A query is a language that describes the retrieval of data from database. As far as query optimization is concerned, it is often assumed that queries are expressed in a content-based manner, giving the optimizer sufficient choices among alternative evaluation procedures. Queries can be expressed in several ways. The most important application is that of direct requests by end users who need information about the structure or content of the database [6]. Another application of queries occurs in transactions that change the stored data based on their current value. Query like expressions can be used internally in a DBMS, to check access rights, maintain integrity constraints, and synchronize concurrent accesses correctly [6].

For Example, SQL query to view employee_no,
employee _name and
salary of those students who have salary more than 30000
from the employee database would be as –

```
SELECT employee _no, employee_name, salary
FROM employee
WHERE salary>30000
```

B. Optimization Objectives

Objective of optimization process should be either to maximize the output for a given number of resources or to minimize the resource usage for a given output.[6] Query optimizer tries to determine the most efficient way to execute a query. A query can be as simpler as finding the “name of person who lives in north street Bombay” or as complex as “finding the name of the person whose age is between 40 to 50 and works as clerk in government sectors and average salary of employee is not more than 25000”.So, query optimization play an vital role to execute such type of query and manipulate them to acquire the relevant output from the database. One can calculate the response time by query optimization and the response time can be minimized by considering following three costs:

i) Transmission
Cost: Cost of transmitting data stored on original site to site where manipulation takes place.
ii) Storage Access Cost: Cost of transferring data from secondary memory to main memory.
iii)CPU time: Cost of computation taken by CPU.

C. Types of Optimization Approaches

i) Query Using Graph

Query Graphs are used in query optimization for the representation of queries or query evaluation strategies. Two classes of graphs can be distinguished: object graphs and operator graphs [7].Nodes in object graphs represent objects such as variables and constants. Edges describe predicates that these objects are to fulfill [8]. Operator graphs describe an operator-controlled data flow by representing operators as nodes that are connected by edges indicating the direction of data movement [9]. Query graphs have many attractive properties. The visual presentation of a query contributes to an easier understanding of its structural characteristics.[6]

Example 1: Consider two relational schemas.

Students (Rollno, Name, Address) and
Student_Marks (Rollno, DBMS, DFS, C,C++, TotalMarks,CPI,SPI)

Assume that the user wishes to view

“Name, TotalMarks ,CPI and SPI" of the students who have got TotalMarks greater than 300.”
The query for this would be:

SELECT Name, TotalMarks,
FROM Students, Student_Marks
WHERE Students.Rollno=Student_Marks.Rollno and
(SPI>8)
The query graph for this query is shown in fig 3.1.

ii) Query Using Tables

Tableaus notations for a subset of relational calculus queries are characterized by containing only AND-connected terms and no universal quantifiers [10].By continuing above example, the queries can be represented as shown in fig 3.2.Tabular format is used to represent the quires as it is easily understandable. The main advantage of tabular representation of relation is the value of any data element can be retrieved by the intersection of rows and columns. The row and column representation of relation helps user to identify the required record easily.

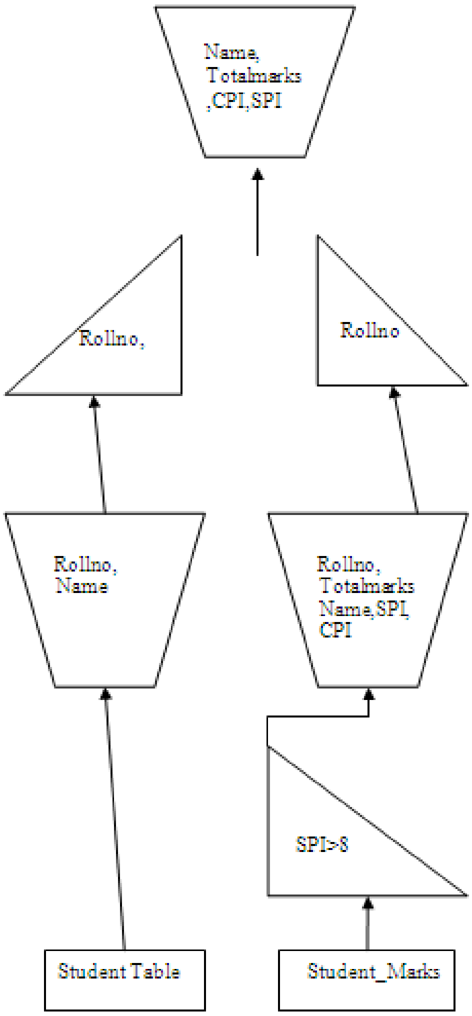


Figure.3.1 Query Using Graph

Table 3.1 T(Student)

Roll No	Name	Total	Percentage
1	S1		
2	S2		
3	S3		
4	S4		

Table 3.2 T(Marks)

Roll No	Name	Total	CPI	SPI
1		T1	CP1	SP1
2		T2	CP2	SP2
3		T3	CP3	SP3
4		T4	CP4	SP4

Table 3.3 T(σTotal>300)

Roll No	Name	Total	CPI	SPI
1		308	CP1	SP1
2		315	CP2	SP2

Table 3.4
T (Join (Students.Rollno =Marks.Rollno) σ Total>300)

Roll No	Name	Total	CPI	SPI
1 1	S1	308	CP1	SP1
2 2	S2	315	CP2	SP2

IV. ACKNOWLEDGMENT

I would like to take this opportunity to express my profound gratitude and deep regard to our Institute, for the guidance, valuable feedback and constant encouragement. I would also like to give my sincere gratitude to our management, all the friends and colleagues who filled in the survey, without which this research would be incomplete. I am also very thankful to my husband Mr. Anant B Patel for all his support and help.

V. CONCLUSION

This Paper shows the different techniques for the representation of query. In this review paper we have analyzed the graph based query representation and table based query representation. Both the ways are easy to understand but query based representation is more easy to understand.Moreever it is effeciaent.There are many other ways for query optimization.

In future the two or more techniques can be combined and the new hybrid method can be developed and implemented.

REFERENCES

[1] Matthias Jarke, Jijrgen KOCH "Query Optimization in Database Systems" Graduate School of Business Administration, New York Uniuersity, New York, New York 10006 , Fachbereich Znformatik, Johann Wolfgang Goethe-Universitiit, 6000 Frankfurt 1, West Germany

[2] Darter C J (1995), Introduction to Database System, Addison Wesley, New York. Tara N. Sainath, Carolina Parada "Convolutional Neural Networks for Small-footprint Keyword SpottingGoogle, Inc. New York, NY, U.S.A"

[3] Dr. (Mrs Pushpa Suri)1 and Mrs Meenakshi Sharma," A Comparative Study Between The Performance Of Relational & Object Oriented Database In Data Warehousing", University, Kurukshetra, India , Colllege of Technology and Management Kaithal, India

[4] Abdullah Dilsat"Query Optimization in Distributed Databases. Report, Middle East Technical University

[5] Sukheja Deepak and Umesh Kumar Singh : A Novel Approach of Query Optimization for Distributed Database Systems. IJCSI International Journal of Computer Science

[6] Jyoti Mor,Indu Kashyap,R. K. Rathy, CSE Department MRIU, Faridabad

[7] Leee Chiang, Chih Chi-Sheng and Chen Yaw-Huei "Optimizing large join queries using a graph-based approach.", IEEE Trans on Knowledge and Data Eng.2001,13(2): p-298-315, 2001.

[8] Youssefi, K. and E. Wong, "Query processing in a relational database management system" In Proceedings of the 5th International Conference on Very Large Data Bases (Rio de Janeiro, Oct. 3-5). IEEE, New York, 1979, pp. 409-417.

[9] Yao S.B, "Optimization of query evaluation algorithms.ACM Trans. Database Syst" Vol 4, 2 (June), p-133-155,1979.

[10] Aho, A.V., Sagiv,Y. and J. D. Ullman: "Efficient optimization of a class of relational expressions" ACM Trans. Database Systems. 4, 4, p- 435-454, 1979.