ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 3, May2012

EFFICIENT KEYWORD SEARCH IN RELATIONAL DATABASES

Navneet Kaur, Rajdeep Kaur , Navjot kaur
Lovely Professional University
Phagwara, Punjab

Abstract- A user who wants to get knowledge from a relational database that needs to know about structured query languages and database schema. Mostly users are not know to those things, so searching knowledge from relational databases is difficult to them. Where a keyword query input is a simple search model that can be issued by writing a list of keywords values, keyword search that place provide a solution of the problem. Because a keyword input query can be interpreted variously, a large number of outputs are returned. And indexing helps to easily retrieved answers and with the help of indexing we measure the performance of the CPU, execution time and Disk memory consumed.
Index Terms- Relational databases, Keyword search, Indexing, Ranking method.

I INTRODUCTION

Every organization has data that needs to managed, analyzed and collected. A relational database system completes these needs. Along with these features of a relational database system come requirement for maintaining and developing the database. Database administrators, data analysts, and database designers need to be able to convert the data in a database into useful data for both day-to-day long-term planning and operations. RDBMS a database system made up of various files with data elements in two-dimensional array. It has the capability to recalculate data elements to from various relations resulting in a very great flexibility of data usage. Relational database management system is a DBMS in which data is saved in tables and the relationships among the data are saved in tables. The data can be reassembled and accessed in many different ways without change the table forms. It is a program that lets we administer, create, and update a relational database. Most commercial relational database management system uses the (SQL) Structured Query Language to access the database, although SQL was changed after the development of the relational database model and is not necessary for its use. Another important feature of relational database systems is that a each single database can be spread across different tables. Relational database differs from flat-file databases, in which every database is self-contained in a single table.
Database technology is growing is day by day. While a rapid growth in the number of users who require to access online databases without having a overall knowledge of the query languages and schema. Keyword search was put forward to solve this problem. Researcher`s wishes their work can make people search semi-structured and structured data just like using web searching engines.

Information Retrieval

Information retrieval (IR) is the area of study related with searching for keyword, for information related keyword, and metadata about keyword, as well as that of searching relational databases, World Wide Web, and the structured storage. There is repetition in the usage of the terms information retrieval, text retrieval, and keyword retrieval, but also has its own body of praxis, theory, literature, and technologies. Information retrieval is interdisciplinary, based on mathematics, library science, information science, computer science, linguistics, cognitive psychology, law and statistics. Automated IR systems are used to eliminate what has been called IO (Information Overload). Many public libraries and universities use information retrieval (IR) systems to provide access to journals, books and other related documents. Applications of IR are Web search engines.

The rest of paper is organized as follows. Section II describe the releated work where we study the idea how we search the keyword. Section III describes the concept of keyword search, keyword search difficulty and ranking method. Section IV describe the architecture of keyword search and in which show the result of indexing in the form of tables and graphs. Section V describes the result of keyword search length analysis according to the user enter query length. Section VI describes how to search the entire database in the database schema. Section VII describes how to search all columns of all tables in the database for

the keyword search and Section VIII include the conclusion and future work.

II RELATED WORK

Guoliang Li & et. al (2009), "Progressive Keyword Search in Relational Databases". In this research Paper database research information retrieval community has recently recognized the benefits of keyword search, we introduce keyword search benefits into relational databases [1].

Lu Qin & et. al (2009), "Scalable Keyword Search on Large Data Streams" In this research Paper is widely realized that the integration of IR (information retrieval) and different database techniques provides users with a wide range of high quality services. We significantly eliminate the number of intermediate outputs when processing joins over a data stream, and therefore can obtain high efficiency [4].

Utharn Buranasaksee & et. al (2010), "Answer Aggregation for Keyword Search over Relational Databases"..In this paper formulate an answer aggregation of keyword search over relational databases problem which merges related joining tuples from multiple tables to a single tuple to reduce redundancy in the results and improve the search quality [6].

III KEYWORD SEARCH

Database research community; start to introduce capabilities of keyword search to relational databases. Two approaches of keyword search are Steiner tree and candidate network. The candidate network find outputs composed of related tuples by extending and generating a candidate network following the primary and foreign key relationship. The Steiner tree model the tuples in relational database schema as a graph form, where edges are primary and foreign key relationship and nodes are tuples.

A Keyword Search Difficulty

In order to we know which keywords to target now, it's necessary to not only understand the demand for a given phrase or term, but the work needed to obtain those rankings. If mighty competitors block the top 10 results and we're just starting out on the web engine, the uphill battle for rankings record can take months or years of support bearing little to no fruit. This is why it's necessary to understand keyword difficulty.

B What Is Ranking Method?

We propose evaluating ranking overall design of database for keyword query to obtain if any are better output to this context than pivoted normalization ranked. Even if these ranked functions do not determine improve search quality, an alternate ranked scheme might allow optimizations in query processing. These optimizations could eliminate execution time without sacrificing search quality.

Another term, including node size and prestige, compactness of outputs, can be added to the ranked function. Although we doubt the existence of a "best" ranking function, we should do thorough experiments with the wealth of ranking operations described in the IR (Information retrieval) literature.

IV ARCHITECTURE FOR KEYWORD SEARCH

In this part the data model, architecture for Keyword search method is discussed in detail and also discussed certain keyword queries.

- Database And Keyword Query Model

Database systems can be classified into different models. Now days, different data model are there, but we use relational database model. Relational database model define a collection of tables which contain all data.

4.1 SYSTEM ARCHITECTURE

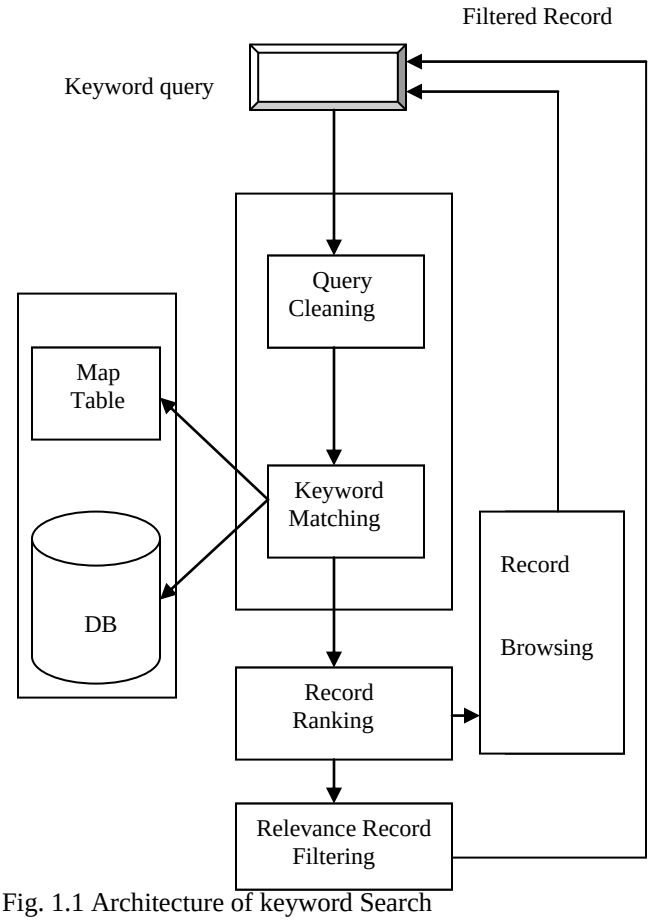

Fig. 1.1 Architecture of keyword Search

4.2 PROPOSED ARCHITECTURE (THE PROPOSED SYSTEM WORKS IN FIVE PHASES)

A Indexing the keyword

Index is one way to access the data quickly. Indexes can be created on any relation of attributes. Queries that filter using those attributes can obtain related tuples randomly using the index, without having to check every tuple in turn. Index is analogous to using the index of a book to go directly to the every page on which the data we are looking for is found i.e. we do not have to read the complete book to find what we are looking for. Relational databases systems typically supply various indexing techniques, each one of which is optimal for some combination of relation size, typical access pattern, and data distribution. Indexes are usually not part of the database, as they are considered a detailed implementation and indexes are usually organized by the similar group that maintains the another parts of the database. It should be noted that efficient indexes use on primary and foreign keys can dramatically improve the query performance. Because number of tuples in a table and hash indexes result are constant time queries. In which we different techniques of used for analyze the performance.

- *Non- Clustered Index*

In which data is present in the form of arbitrary order. In which logical order of data is present by the index. Data rows are divided throughout the table regardless of the value of indexed column or expression. Non-clustered index have index keys in sequence order, with the leaf node of the index that access the pointer to the record.
In a non-clustered index include:
1. The physical order of the rows is not same as the index order.
2. Basically organized on non-primary key columns that is used in WHERE, and JOIN clauses. It can be more non-clustered index on a relational database table.

- *Clustered Index*

Cluster change the data block into a different order to match the index, output in the row data being that is stored in order. In which only one clustered index can be created on a database table. Clustered indexes table can maximize whole speed of retrieval, but usually where the data is obtained sequentially in the similar or opposite order of the clustered index. The Physical records are in particular sequence order on various disks, the next row data item in the order is fastly before or after the last one. The primary property of a clustered index is sequence of the physical data rows item in order with the index blocks item that point to them. Many databases separate the data and different index blocks into separate files, other blocks put two completely separate data blocks within the same file. We create that object where the physical sequence of rows data is similar as the index sequence of the rows data and the leaf level of clustered index obtain the actual data rows.

- *Full Text Index*

In SQL server, full text search is obtained using Full text indexing. It provides full text input queries opposite character based data. Searches can include multiple forms of a phrase/word, phrase, words etc. We create full text indexes for columns referenced in the queries. It is a made oh word and phrase tokens derived from the indexed text.

- *INDEXING RESULTS*

a) Execution Time Consumed Analysis

When we search record in the database then we need to a dataset. Firstly, we using that dataset we analyze the CPU, execution time and memory consumed. This is all doing with the help of different indexing techniques Non-Clustered, Clustered, and Full-text Index. We easily analyze the whole system performance. In Fig.1.2 and table 1.1 describe, when we use the dataset that is containing records 50,000 to 4,50,000. Based on this dataset we analyze execution time for Non-Clustered, Clustered, and Full Text index. In which when we have 50,000 then execution time for Non-Clustered Index is 322, Clustered Index for 387, and Full Text Index for 220. With the help of we analyze how much execution time for records. In the table and graphs we show different records and they take different execution time. We easily analyze which records for execution time minimum.

Table 1.1 Records for Execution Time Analysis

Records	Non Clustered Index	Clustered Index	Full Text Index
50,000	322	387	220
1,00,000	776	789	387
1,50,000	1256	2787	587
2,00,000	1879	3773	867
2,50,000	2278	4967	976
3,00,000	2843	5867	1289
3,50,000	3523	7409	1489
4,00,000	4023	7798	1787
4,50,000	4684	8967	1867

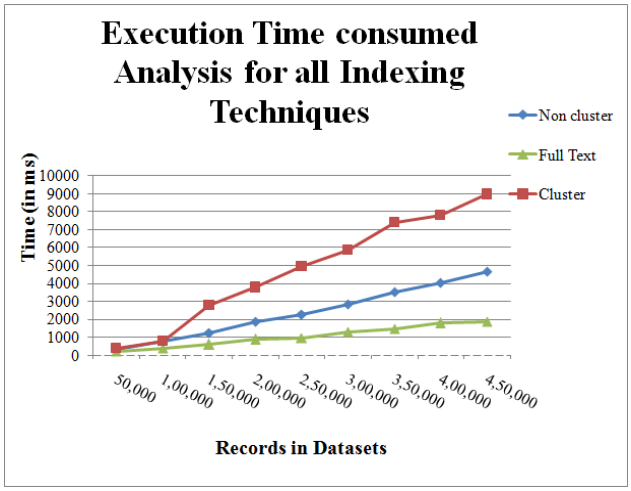

Fig.1.2 Execution Time Consumed

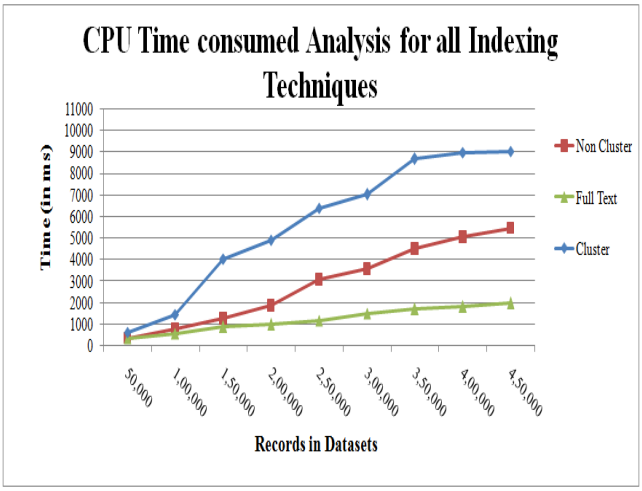

Fig.1.3 CPU Time Consumed

b) CPU Time Consumed Analysis

In which we used the same dataset that is used for execution time analysis. In CPU time analysis when we used record 1,00,000 then CPU time consumed for Non-Clustered Index is 756, Clustered Index for 1434, and Full Text Index for 587. In which we take different record and analyze different CPU time consumed. With the help of this we analyze which one records according to our requirement is best that is take minimum CPU time consumed.

Table 1.2 Records CPU Time Analysis

Records	Non Clustered Index	Clustered Index	Full Text Index
50,000	344	643	367
1,00,000	756	1434	587
1,50,000	1254	4023	867
2,00,000	1864	4922	987
2,50,000	3076	6409	1172
3,00,000	3560	7023	1489
3,50,000	4489	8678	1698
4,00,000	5045	8992	1798
4,50,000	5467	9034	1967

c) Disk Memory Consumed Analysis

In which Disk Memory Consumption for we use different records and we analyze the outputs for different records. When we used record 3,50,000 then Disk Memory Consumption for for Non-Clustered Index is 34782, Clustered Index for 420, and Full Text Index for 28673. Then we analyze for 3,50,000 records clustered index take less memory. And with the help of this we easily analyze which one records is best according to our requirement that is used minimum CPU time consumed.

Table 1.3 Records for Disk Memory Consumed

Records	Non Clustered Index	Clustered Index	Full Text Index
50,000	5033	109	5289
1,00,000	11782	130	10872
1,50,000	14897	208	15892
2,00,000	19672	263	16892
2,50,000	24762	290	25892
3,00,000	29673	373	24873
3,50,000	34782	420	28673
4,00,000	38873	480	29733
4,50,000	44893	529	50332

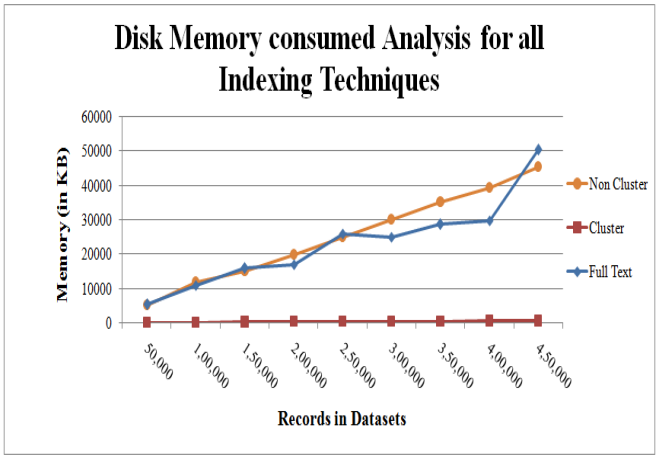

Fig.1.4 Disk Memory Consumed Analysis

B Searching the keyword query

In which we two terms query include cleaning and keyword matching. When user enters a query then spelling corrections and semantic linkage is checked and then output is a cleaned query. And keyword search , in which user enter a input query that input query search in the database schema then find a output.

C Ranking the result

When user enter a keyword query as a input in the SQL search engine, then user have many different options to various tuples returned as a output query answer. For finding relevant answer we rank the whole database on the basis of particular query number of time present in the database and query length. The ranking method is best for obtain the relevant answer.

$$\text{Score}(P_i, Q) = \sum W_k / Q_t$$

Where W_k= total no. of keyword present in the database with respect to the user input keyword query.

Q_t = total no. of keyword present in the user input query(length of the query).

D Filtering relevance record

When user search keyword query in the search engine then we define the threshold value, according to threshold value we filtering the relevance record. When answer match with the given keyword query then we find the output.

E Record Browsing

When system find the related information with the answered tuple set , then answer set needs to be designed in the form of schema graph. A schema graph include all the relational database tables, and the relationship between the tables.

V RESULT OF KEYWORD SEARCH QUERY LENGTH ANALYSIS

In which we describe according to the query length how much time to take searching record in the database. In above given Fig 4.4 and Table 4.4 we search record nitu that keyword length is 4 and that take time 156 ms for search record in the database. And in which we easily analyze for which record how much time to search data in the database.

Table 1.4 Records for Keyword Search Analysis

Search name	Time Consumed
Nav	140
Nitu	156
Lilly	200
Navjot	300
Pardeep	320
Rupinder	380

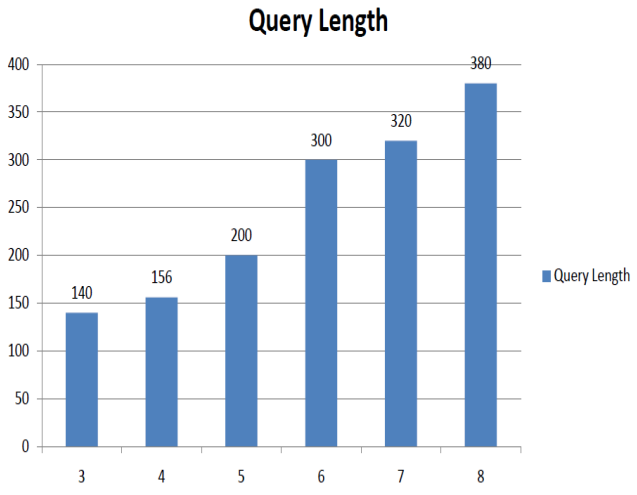

Fig.1.5 Keyword Search Query Length Analysis

VI HOW TO SEARCH THE ENTIRE DATABASE

Microsoft Operations Manager (MOM), uses the SQL Server for saved the data all different computer, performance and alert related knowledgeable data. We narrowed the problem down to something needed a script that can find all different MOM tables for a particular string. We had no such script, so we ended up finding data manually. That's when I really felt the requirement that script and came up with that stored procedure "SearchAllTables". It obtain a string as input query parameter goes and find all varchar, char, nvarchar, nvchar columns of all different tables, owned by all different users in the current relational database.

VII HOW TO SEARCH ALL COLUMNS OF ALL TABLES IN A DATABASE FOR A KEYWORD

MOM (Microsoft Operations Manager) really felt the need for such a record and came up with this particular stored Procedure "SearchAllTables". It select a search keyword as input query parameter goes and find char, varchar, nchar, and nvarchar columns of all different tables, owned by all different users in the current relational database. This procedure free to extend find other different datatypes.
Results of this stored procedure contain two different columns:

- The table and column name in which the answer keyword was found.
- The actual content of the column.

Here`s a word of problem, before we find and start this procedure. This procedure is very easily on smaller databases, it could take one or more hours to complete, on a huge database with too much large character columns and a large number of rows. If we are trying to run it on a huge database, be prepared to wait. It is necessary to use Full-Text index find feature for free text finding, but it doesn`t make sure for this kind of particular ad-hoc requirements.

This stored procedure is create in the required database. And below how we run:

To search all columns of all tables in management database for the keyword "Neetu"

```
EXEC SearchAllTables 'Neetu'
GO
```

VIII CONCLUSION

In this paper, firstly we indexing the whole database and find the performance of CPU, execution time, and disk memory consumption. After we proposed a new method ranking that is best for finding relevant answer from a relational database schema. With the help of graph results we examine how much time take to search according to keyword input query length. In future, we work upon the keyword search in distributed environment.

ACKNOWLEDGEMENT

Foremost, I would like to express my sincere gratitude to Ms. Rajdeep Kaur who gave her heart whelming full support in the completion of this research paper with her stimulating suggestions and encouragement to go ahead in all the time of the research paper. She has always been a source of inspiration and confidence for me. She has beaconed light to me as a guide at all stages of preparation of my research paper. At last I am very thankful to my GOD who has given me this golden opportunity to do M.Tech as well as to do research work.

REFERENCES

[1] Guoliang Li, Xiaofang Zhou, Jianhua Feng, Jianyong Wang (2009)†" *Progressive Keyword Search in Relational Databases*" IEEE (2009).
[2] Phyo Thu Thu Khine, Htwe Pa Pa Win, Khin New Ni Tun (2011), "*Efficient Relational Keyword Search System*".
[3]Ian De Felipe, Vagelis Hristidis Naphtali Rishe(2011), "*Keyword Search on Spatial Databases*", IEEE, 2011
[4] Lu Qin, Jeffrey Xu Yu, Lijun Chang, Yufei Tao "*Scalable Keyword Search on Large Data Streams*".
[5] Liang Jeff Chen, Yannis Papakonstantin(2010), "*Supporting Top-K Keyword Search in XML Databases*".
[6] Utharn Buranasaksee, Kriengkrai Porkaew, and Umaporn Supasitthimethee (2010), "*Answer Aggregation for Keyword Search over Relational Databases*".
[7] Bolin Ding, Bo Zhao, Cindy Xide Lin, Jiawei Han(2011), "*Efficient Keyword-Based Search for Top-K Cells in Text Cube*".
[8] Roberto De Virgilio (2011), "*Efficient and Effective Ranking in Top-k exploration for Keyword Search on RDF*".
[9] Zhancheng kong & kunlong zhang (2011), "*Summarizing keyword search in relational database".*

AUTHOR PROFILE

Navneet Kaur, currently doing M.Tech from Lovely Professional University (Phagwara). Research paper published "A Review Study of Keyword search over Relational database using ranking method" in ICRTCTA, April(2012). Currently doing research "Keyword Search in Distributed Environment".

Rajdeep Kaur, currently work as a Assistant Professor in Lovely Professional University(Phagwara). She has done M.Tech from Lovely Professional University. Now a days, doing research on data mining algorithms.

Navjot Kaur, doing M.Tech from Lovely Professional University (Phagwara) in information technology. Currently research on "Enchancement in K-Means Clustering Algorthims using ranking method" and Paper published in "Comparsion of K-Means clutersing and various clustering algorithms", ICRTCTA, April (2012).