Available online at http://www.journalijdr.com

IJDR

International Journal of DEVELOPMENT RESEARCH

ISSN: 2230-9926

International Journal of Development Research
Vol. 6, Issue, 03, pp. 7100-7105, March, 2016

Full Length Research Article

AN EFFICIENT AND IMPROVED PERFORMANCE FOR KEYWORD SEARCHING IN LARGER DATABASES

[1]Arulmurugan, A., [2,*]Nandini, R., [3]Jayasri, P., [4]Rahini, E. and [5]Priyanka, B.

[1]Associate Professor Department of Information Technology, A.V.C College of Engineering, Mannampandal
[2]B.Tech - Information Technology,Department of Information Technology, A.V.C College of Engineering, Mannampandal

ARTICLE INFO

Article History:
Received 29th December, 2015
Received in revised form
18th January, 2016
Accepted 25th February, 2016
Published online 31st March, 2016

Key Words:
Keyword Search,
Top-Notch,
Information Retrieval,
Ranking,
Relational Database,
Keyword Query,
Co-Related Databases.

ABSTRACT

Keyword Search is an emerging field and other way of dealing with traditional SQL querying in relational database with larger datasets. Keyword search as possible in co-related database simply you can type in words that describe your research topic and retrieve records containing those search terms. Keyword search enable user to produce result in easy way does not having a detailed knowledge about either schema or complex query languages. At present work on single database use keyword query search but user need to access more than one database. To overcome the problem of solution in advance system on larger database discover information based on user needs and an efficient keyword search algorithm in co-related databases using new novel is DeINIX (Density Inverted IndeX) using to reduce memory space which will clearly show the actual result also it displays keyword search (top-notch) with ranking top-of-list-10 and Final outcome of this method answers queries more accurately, memory space to retrieve a quality answer and execution time is less.

INTRODUCTION

A database is a place of storing information from that required information can be retrieved. Relation database is the one where information and multiple dataset. There are represent in tables and records by means of rows and columns. Presently keyword search handle with single databases. Keyword Search is latest technique in database search. User simply inserts a keyword for looking out and gets a result associated with that keyword. Keyword search on relational dataset find the solution of the tuples which are connected to database keys like primary key and foreign keys. There fore these method additionally present which comparative techniques used for keyword search like DISCOVER, DBXPLORER and BANKS. Keyword search alter the user to access the database in real time while not knowing of SQL or database schema.

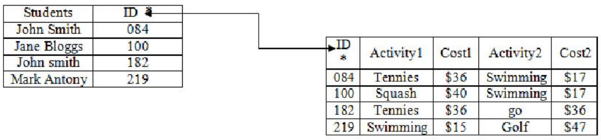
Example 1: We are taking the sample student database as shown in figure 1 as details of SPORTS; it has two table as STUDENTS table and ACTIVITY table ACTIVITY table which gives activities details and cost details. STUDENT table gives student detail. In case STAFF of a corresponding Institution wants to know Particular students activity he has to simply write the keyword. i.e., students id it will retrieve the corresponding details from ACTIVITY table. The answer was obtained by joining the stud-id(STUDENT)= id (ACTIVITY) ; through foreign key join but this will a complex task doing in SQL based queries and it is more useful compared to SQL.

Students Table

Example 2: The above example shows Figure 2 as EDUCATION department. In case a has to send student “John” to extend the activity for tennis So what the admin does it take the PARTICIPANTs DATABASE TABLE (figure 2a) and STUDENT DATABASE TABLE (figure 2b) he use the keyword john and tennis from the two databases. The query answer shows whether john well trained with tennis.

**Corresponding author: Nandini, R.,*
B. Tech - Information Technology,Department of Information Technology, A.V.C College of Engineering, Mannampandal.

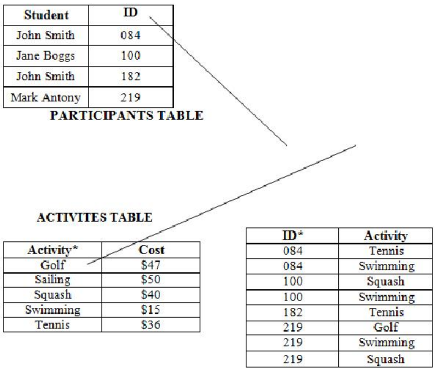
The previous example shows that the keyword search is workable over co-related databases. But we search in more than one database there are so many practical setting we need to do so that we can have an efficient search in co-related database.



Students	ID *
John Smith	084
Jane Bloggs	100
John smith	182
Mark Antony	219

ID *	Activity1	Cost1	Activity2	Cost2
084	Tennies	$36	Swimming	$17
100	Squash	$40	Swimming	$17
182	Tennies	$36	go	$36
219	Swimming	$15	Golf	$47

Figure 1. Sample database of Student Details



Student	ID
John Smith	084
Jane Boggs	100
John Smith	182
Mark Antony	219

PARTICIPANTS TABLE

ACTIVITES TABLE

Activity*	Cost
Golf	$47
Sailing	$50
Squash	$40
Swimming	$15
Tennis	$36

ID*	Activity
084	Tennis
084	Swimming
100	Squash
100	Swimming
182	Tennis
219	Golf
219	Swimming
219	Squash

Figure 2. Keyword Search over Co-Related databases

In our proposed system an algorithm of searching in Co-Related Databases for efficient search and also using method is DeINIX (Density Inverted IndeX) a novel inverted index Keyword search use a variety of techniques and algorithm used for storing and retrieving information, less accuracy, but it doesn't giving an accurate answer, need massive time for looking for a particular queries and need great amount of space for storing for information storage.

EXISTING SYSTEM

Many researches are proposed for keyword search in single databases and [3] existing approaches is split into two major techniques. At the start technique in one half is Schema based approach is considered a graph wherever node represents data relationships and edge described foreign keys to primary keys dependencies. Operating of this technique is as follow to begin with the relationship that contains keyword and graph is taken as input then totally possible path is discovered. Second half appropriate queries retrieved because the tuple under this techniques as DBXPLORER [Chaudhuri and Das, 2002] and DISCOVER [VagelisHristidis and YannisPapakonstantinou, 2002] each algorithm uses candidate network to join the tuple and rank them.

Another techniqueis that the Tuple based approach the data graph is contain node. This indicating tuple identifier and an edge act for the foreign key mentioning among two tuples. These techniques perform in single stage is that the scheme search the graph and construct the tree and connection the tuples. BANKS [Bhalotia *et al*., 2002] decrease support this types; it challenge stretch network heuristic by execution single prelusion shortest approach. It uses best initial search strategy around theedge in backward direction. Schema base is useda lot of on deal with schemas are simple instead of tuples. The algorithm proposed in nowadays deal within single database, our paper we'll try and focuses on more than one databases.

LITERATURE SURVEY

Fang Liu, Clement Yu ,*et al* proposed a new concept in keyword search of Relational database contain more text data is expanding very quickly and general users to search information is radically going up and full-text search as possible in RDBMs. User search require information to make clear knowledge of the database schemas and structured query language. This method of searching performance is more problematical for normal user. Outstanding success of information retrieval (IR) form keyword search on the web.

Keyword search in relational databases has latterly developed as a fresh research matter. The specified keyword query is handle in three steps (1) The technique produce each and every answers for the query.(2) The technique work out a ranking score for all answer and also ranks them.(3) In conclusion, top-k answers are returned. ArvindHulgeri, Gaurav Bhalotia, *et al* proposing concept, In first present a examine work on keyword querying in databases. After which we have developed to report on the BANKS system. BANKS put together keyword querying and communicating browsing of databases. By nature, keyword queries are vague, and we need a model for resulting keyword queries. BANKS, like an existing system called DataSpot, and a database as a graph. In this model, tuples resemble to nodes and foreign key and other links betwixt tuples resemble to edges. Outcome to a query are modeled as rooted trees connecting tuples that match individual keywords in the query.

Answers are ranked using a notion of proximity coupled with a notion of prestige of nodes and latter techniques using developed for Web search. One advantage as our model as possible of several extensions A keyword in a tuple/relation-name may not be rightly equal to a search term but alternatively be a signification to it.Functioning issues caused by metadata keywords matching large numbers of nodes are being addressed in BANKS. V. Hristidis and Y. Papakonstantinou, DISCOVER deliver qualified connected networks of tuples, that is, sets of tuples that are associated because they join on their primary and foreign keys and totally contain all the keywords of the query. In initial stage the Candidate Network generator produce all candidate networks of relations; Plan Generator builds plans for the powerful evaluation of the group of candidate networks. Using Greedy algorithm return a near optimal execution plan based on the actual cost model. Some advantage a huge amount of information is stored in relational databases but not supported to discover. The user of DISCOVER does not need knowledge about schema, SQL and roles and terms used in query. Searching data based on the relational keywords. One drawback as time complexity is high.H. He, H. Wang, J. Yang, and P.S. Yu concept advantage as ranking is applied to each and every Search and problem is unused memory becomes large, so huge memory is wasted. G. Li, J. Feng, X. Zhou, and

J. Wang benefit Keyword based searching mechanism is presented internally. Demerit is hard to manipulate large set of data presented in database.

PROBLEM STATEMENT

The problem describe of the keyword search over co-related database. Keyword search for the single database [7] can be done by LIKE predicate which is already predefined in SQL and also by creating a User-Defined Function in SQL this two method which works without index. Keyword search in single database require two methods are No-index Method and Index Based Method. We must know the underlying schema and instances to search, which is the problem in traditional database the complexity to fire a query increases. Earlier works uses dual approach as Schema based approach and Tuple Based approach. These types of approach works in only unique phase, among these two concept schemas are easy rather than tuples and are less space consuming but it takes more time as it need to scan all the document in table. The difficulty of top-k processing [Ihab *et al*., 2008] has attracted in recent studies.

The approach developed in this system is Search-as-you-type in Co-Related databases. In this proposed technique the link is created between two databases by connecting foreign key between the databases. Existing approach we have to create Prefix table, Inverted Index table and keyword table which take more time and space for execution. In technique added a new method DeINIX which occupies less memory space compare to existing algorithm and which can reduce the pre-processing time with the help of this technique.

SEARCHING FOR SINGLE DATABASE

In single database [Guoliang Li *et al*., 2013] searching for single keyword queries and database to support search-as-you-type is done using SQL by two different methods are No-index Methodand Index Based Method.

First, No-index Method one section is Using Like predicates: In SQL there is an in-built predicate LIKE which allows matching the keyword and retrieving that record. Second section is Using User-Defined Function (UDF): We can add the functions into the databases to examine whether a record contains the query keyword or not. Second, Index Based Method a database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and storage space to maintain the index data structure. Indexes are used to quickly locate data without having to search every row in a database table every time a database table is accessed. Indexes can be created using one or more columns of a database table, providing the basis for both rapid random lookups and efficient access of ordered records.For making an inverted index table, firstly an single id is assign to keywords in alphabetic order which are in Table T. Afterwards inverted index table IT is created with records in form<key id,rec id> where key id is the id of keyword and rec id is id of record For a table T, Prefix table PT is created with <prefix, skey id, lkey id> where p represent the prefix of keywords, lkey id the largest string id of prefix p and skey id smallest string id of p. An example; TABLE II of inverted index is created from Table 1. The inverted index has tuple <key 10, rec 4> the key 10 has keyword "VLDBJ". The prefix table constructed is ("v", key 9, key 10), now key 9 has "VLDB" with minimum length and k10 has "VLDBJ" with maximum length. Hence id will in range < key 9, key 10 >. For a given Partial Keyword w, we first extract the range of prefix table PT then find the record in the following range which is shown in figure 3.

The SQL query for is written as below:

SELECT T.* FROM PT, IT, T WHERE PT.PREFIX ='W'AND PT. skey id ≥ IT.key id AND IT.rec id=T.rec id. [7]

With the above SQL we can retrieve the same for "v"

SELECT dblp.* FROM Pdblp, Idblp, dblp WHERE Pdblp. prefix ='v' AND Pdblp. ukid ≥Idblp. Key id AND Idblp.rec id=dblp.rec id.[7]

Drawback on single database more over the Keyword Table, Inverted Index and Prefix table which are created occupy more space and also preprocessing time too. SQL, for that we need to learn an underlying schema and the SQL language to retrieve a Keyword Query Search. In our proposed an efficient algorithm of Searching in co-related Databases by DeINIX which will answer the keyword queries without knowing the underlying schema language. Advantage is that it takes less time for execution and memory utilization is less.

PROPOSED SYSTEM

This system describes the general architecture and workflow of the PROPOSED work is shown in Figure 4. The architecture consists of the two components are Pre-Processing and Query Processing. Both processing are explained in detail.

DeINIX: Density INvertedIndeX

In classic Inverted Index approach, keywords appearing in the database have been assigning different IDS (TABLE III). Let me explain by an example in Table III the word "schema" is occurring in rec 1, rec 2, rec 3 and rec 4these index are stored specifically in Inverted Index table (Table II 3(b)) which result in more memory utilization and hence we propose a method DeINIX to support keyword search using range list and range intervals <rec 1, rec 4> which reduces the memory space of the Inverted Index table and creating performance better and fast. Again a question arises what if a we have keyword like "VLDB" is occurring only once in database, solution for ie. We have to write < rec 4, rec 4> for occurrence of the keyword once. Other condition is if we can't have continuous interval then we can write as < rec 3, rec 3>, < rec 6, rec 6,> and < rec 8, rec 8> for "ICDE" keyword.

Keyword Matching

The essential idea of keyword matching is to determine the answer for the keyword query. To match the keyword user can type the keyword and receive the result by retrieving from the DeINIX table and also apply Searching in Co-Related Databases.

Table 1. dblp A Sample Publication Table

ID	Title	Authors	Book Title	Year
rec 1	Efficient retrieval of the top-k most relevant spatial web objects	G. Cong, C. S. Jensen, and D. Wu	PVLDB	2009
rec 2	DBXplorer: A System for Keyword-Based Search over Relational Data Bases	S. Agrawal, S. Chaudhuri, and G. Das	ICDE	2002
rec 3	flexible and efficient xml search with complex full-text predicates	S. Amer-Yahia, E. Curtmola, and A. Deutsch	SIGMOD	2006
rec 4	Retrieving top-k prestige-based relevant spatial web objects	X. Cao, G. Cong, and C. S. Jensen	VLDBJ	2002
rec 5	Similarity flooding: a versatile graph matching algorithm and its application to schema matching	S. Melnik, H. Garcia-Molina, and E. Rahm	ICDE	2002
rec 6	Indexing relational database content offline for efficient keyword-based search.	Q. Su and J. Widom	ICDT	2005
rec 7	Peer-to-peer management of xml data: issues and research challenges.	G. Koloniari and E. Pitoura	SIGMOD	2005
rec 8	Discover: keyword search in relational databases	VagelisHristidis and YannisPapakonstantinou	VLDB	2002
rec 9	Keyword Search over Relational Databases: Issues, Approaches and Open Challenges	Sonia Bergamaschi, Francesco Guerra, and Giovanni Simonini	SIGIR	2014
rec 10	keyword searching and browsing in databases using Banks	G. Bhalotia, A. Hulgeri, C. Nakhey, S. Chakrabarti, and S. Sudarshan	ICDE	2002

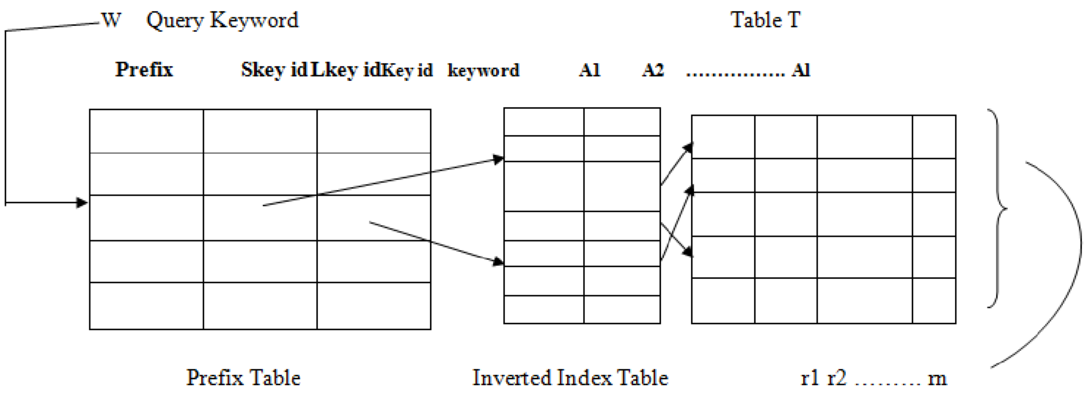
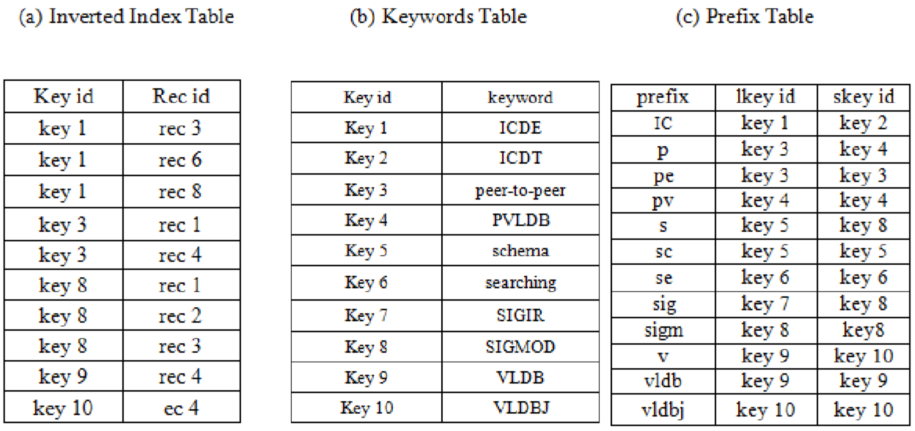


Figure 3. search-as-you-type support on prefix table and inverted index

Table 2. Prefix Table and Inverted Index Table



(a) Inverted Index Table

Key id	Rec id
key 1	rec 3
key 1	rec 6
key 1	rec 8
key 3	rec 1
key 3	rec 4
key 8	rec 1
key 8	rec 2
key 8	rec 3
key 9	rec 4
key 10	ec 4

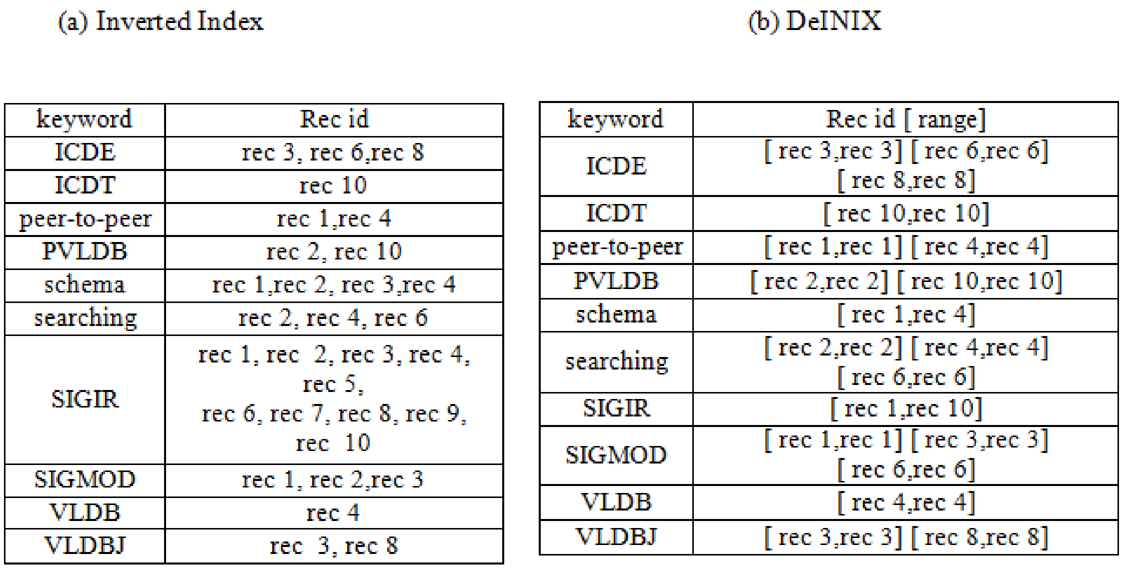
(b) Keywords Table

Key id	keyword
Key 1	ICDE
Key 2	ICDT
Key 3	peer-to-peer
Key 4	PVLDB
Key 5	schema
Key 6	searching
Key 7	SIGIR
Key 8	SIGMOD
Key 9	VLDB
Key 10	VLDBJ

(c) Prefix Table

prefix	lkey id	skey id
IC	key 1	key 2
p	key 3	key 4
pe	key 3	key 3
pv	key 4	key 4
s	key 5	key 8
sc	key 5	key 5
se	key 6	key 6
sig	key 7	key 8
sigm	key 8	key8
v	key 9	key 10
vldb	key 9	key 9
vldbj	key 10	key 10

Table 3. Working of Traditional Inverted Index and DeINIX



(a) Inverted Index

keyword	Rec id
ICDE	rec 3, rec 6,rec 8
ICDT	rec 10
peer-to-peer	rec 1,rec 4
PVLDB	rec 2, rec 10
schema	rec 1,rec 2, rec 3,rec 4
searching	rec 2, rec 4, rec 6
SIGIR	rec 1, rec 2, rec 3, rec 4, rec 5, rec 6, rec 7, rec 8, rec 9, rec 10
SIGMOD	rec 1, rec 2,rec 3
VLDB	rec 4
VLDBJ	rec 3, rec 8

(b) DeINIX

keyword	Rec id [range]
ICDE	[rec 3,rec 3] [rec 6,rec 6] [rec 8,rec 8]
ICDT	[rec 10,rec 10]
peer-to-peer	[rec 1,rec 1] [rec 4,rec 4]
PVLDB	[rec 2,rec 2] [rec 10,rec 10]
schema	[rec 1,rec 4]
searching	[rec 2,rec 2] [rec 4,rec 4] [rec 6,rec 6]
SIGIR	[rec 1,rec 10]
SIGMOD	[rec 1,rec 1] [rec 3,rec 3] [rec 6,rec 6]
VLDB	[rec 4,rec 4]
VLDBJ	[rec 3,rec 3] [rec 8,rec 8]

Searching co-related database

The main task is to process the keyword over co-related databases. In this system algorithm is proposed to link the databases through foreign keys in order to connect the foreign keys. In earliest stage we need to identify of all foreign keys in the individual database. Both databases must be sharing approximate common attribute field after only they will be joinable. The next stage is to link up two databases with support of foreign keys. Now the databases are connect the selection of the keyword is based on DeINIX table. After the searching performance is finished it displays the result and will go to next process.

Top-notch Search

After the query is created, several result tuples are produced. At present the question arises which answer is same to the keyword search for the user. In this system we display only top-notch means first 10 answer from several tuples displayed in the list.

IMAGE RETRIEVAL

Text Based Retrieval

The process of retrieving images from the database is the primary thing. An annotator prepared database in which Images were stored along with string attributes – keywords that reflected in a relatively broad manner the content of the image. Advantage of text-based image retrieval is well-developed information retrieval algorithms as well as good mechanisms

Color Based Retrieval

To classify the images clearly, color is a low-level image feature,. Some CBIR system is only used to utilize color as the image retrieval feature. so that color has advantage for image retrieval. At a single pixel of the image it provides multiple measurements, so that categorization was done without the need for complex spatial decision-making. The process like manipulate and extract of image is done easily.

Conclusion

The actual model has use keyword search for access single database only. In this system can use DeINIX for relational database to less the storage area. By using DeINIX algorithm, answers keyword queries at relational database it provide more efficiently when compared to current system. In proposed technique the link is created between two databases by joining the foreign key between the databases. For the existing approach we should create Prefix table, Inverted Index table and keyword table which take long time and space for carrying out. In future process, an approach can be developed the ranking system for co-related database which will reduce carryout time and we should try for indistinct keyword search. In future, strong security approach can be developed for co-related database; also can try to provide information based on image more efficient and effective

REFERENCES

Bhalotia, G. Hulgeri, A. Nakhe, C. Chakrabarti, *et al*, 2002. “Keyword searching and browsing in databases using BANKS” , in IEEE.

Chaudhuri, S. and Das, G. 2002. “DBXplorer: A System for Keyword-Based Search over Relational Data Bases,” in ICDE .

Guoliang Li, JianhuaFeng and Chen Li 2013.,“Supporting Search-As-You-Type Using SQL in Databases”, in IEEE.

Gurmeet Kaur, Er. Arshdeep Singh”A Review Paper on Content Based Image Retrieval”Department of CSE, AIET, Faridkot, Punjab, India

Ihab F. Ilyas, George Beskales, and Mohamed A. Soliman, 2008. “A Survey ofTop-k Query Processing Techniques in Relational Database System”, ACM.

Sudarshan, S. SoumenChakrabarti, *et al*. 2010. “Keyword Search in databases”, in Springer.

Vagelis Hristidis, YannisPapakonstantinou**,** 2002. **“**Discover: keyword search in relational databases**”,** in VLDB .

****