Indexing in Database

Rajni Rani[1]
Master of Technology in Computer Science & Engineering, Sri Guru Granth Sahib World University, Fatehgarh Sahib, Punjab, India.

Abstract - *Databases can store large amounts of data. each stored entity is a complex structure, called a record. Records are indexed based on the values of certain fields. A techniques for indexing data is provide.a method for compressed an index to obtain a compressed index that is easily stored and transmitted. The invention also provides for the decompression of such a compressed index. The main aim of this paper is to provide to the different techniques of indexing in database.*

Keyboards:- *types of indexing , techniques and methods.*

I.INTRODUCTION

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space. Indices can be created using one or more columns of a database table, providing the basis for both rapid random lookup and efficient access of ordered records.Indexing is a way of sorting a number of records on multiple fields[6]. Creating an index on a field in a table creates another data structure which holds the field value, and pointer to the record it relates to. This index structure is then sorted, allowing Binary Searches to be performed on it. The downside to indexing is that these indexes require additional space on the disk, this file can quickly reach the size limits of the underlying file system if many fields within the same table are indexed.

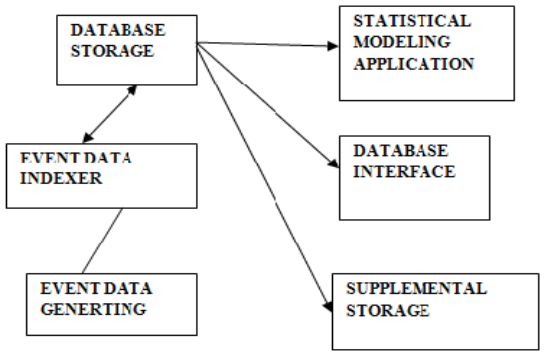

Fig 1.1 Indexing

A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of slower writes and increased storage space. Indices can be created using one or more columns of database table, providing the basis for both rapid random and efficient access of ordered record.

In this paper we did literature survey of indexing in database. The paper gives the overview of various methods of indexig. These various methods have also been discussed. The rest of the paper is organized as below. Section 2 presents the typesof indexing , Section 3 represent the techniques .and Section 4 gives the blocking methods, Section 5 gives the conclusion

II. TYPES OF INDEXING

A. Bitmap index

A bitmap index is a special kind of index that stores the bulk of its data as bit arrays (bitmaps) and answers most queries by performing bitwise logical operations on these bitmaps. The most commonly used indexes, such as B+ trees, are most efficient if the values they index do not repeat or repeat a smaller number of times. In contrast, the bitmap index is designed for cases where the values of a variable repeat very frequently. For such variables, the bitmap index can have a significant performance advantage over the commonly used trees[7].

B. Dense index

A dense index in databases is a file with pairs of keys and pointers for every record in the data file. Every key in this file is associated with a particular pointer to a record in the sorted data file. In clustered indices with duplicate keys, the dense index points to the first record with that key[7].

C. Sparse index

A sparse index in databases is a file with pairs of keys and pointers for every block in the data file. Every key in this file is associated with a particular pointer to the block in the sorted data file. In clustered indices with duplicate keys, the sparse index points to the lowest search key in each block[7].

D. Reverse index

A reverse key index reverses the key value before entering it in the index. E.g., the value 24538 becomes 83542 in the index. Reversing the key value is particularly useful for indexing data such as sequence numbers, where new key values monotonically increase[7].

III. INDEXING TECHNIQUES

A. Sorted Neighbourhood Indexing

This technique was first proposed in the mid 1990s .Its basic idea is to sort the database(s) according to the BKVs, and to sequentially move a window of a fixed number of records w (w > 1) over the sorted values. Candidate record pairs are then generated only from records within a current window[1].

(i) *Sorted array based approach*
(ii) *Inverted index based approach*
(iii) *Adaptive sorted neighbourhood approach*

B. Q-gram Based Indexing

The aim of this technique is to index the database(s) such that records that have a similar, not just the same, BKV will be inserted into the same block. Assuming the BKVs are strings, the basic idea is to create variations for each BKV using q-grams (sub-strings of lengths q), and to insert record identifiers into more than one block.

C. Suffix Array Based Indexing

This technique has recently been proposed as an efficient domain independent approach to multi-source information integration .The basic idea is to insert the BKVs and their suffixes into a suffix array based inverted index. A suffix array contains strings or sequences and their suffixes in an alphabetically sorted order. Indexing based on suffix arrays has successfully been used on both English and Japanese bibliographic database[1].

(i) *Robust Suffix Array Based Indexing*

D. Canopy Clustering

This indexing technique is based on the idea of using a computationally cheap clustering approach to create high-dimensional overlapping clusters, from which blocks of candidate record pairs can then be generated Clusters are created by calculating the similarities between BKVs using measures such as Jaccard or TF-IDF/cosine . Both of these measures are based on tokens, which can be characters, q- grams or words. They can be implemented efficiently using an inverted index which has tokens, rather than the actual BKVs, as index keys[1].

(i) Threshold based approach
(ii) Nearest neighbour based approach

E. String-Map Based Indexing

This indexing technique is based on mapping BKVs (assumed to be strings) to objects in a multi-dimensional Euclidean space, such that the distances between pairs of strings are preserved. Any string similarity measure that is a distance function (such as edit-distance) can be used in the mapping process. Groups of similar strings are then generated by extracting objects in this space that are similar to each other. The approach is based on a modification of the Fast Map algorithm, called String Map, that has a linear complexity in the number of strings to be mapped .[1]

F. Traditional Blocking

This technique has been used in record linkage. All records that have the same binary key value(BKV) are inserted into the same block, and only records within the same block are then compared with each other. Each record is inserted into one block. Blocking has been used in record linkage. All records that have the same BKV are inserted into the same block, and only records within the same block are then compared with each other. Blocking is that errors and variations in the record fields used to generate BKVs will lead to records being inserted into the wrong block. Blocking techniques focusing on accuracy and performance have been proposed to enhance the Entity Resolution process. Blocking is to reduce the pairs of records that need to be examined by partitioning the records into blocks[1].

IV. BLOCKING METHODS

The Standard Blocking method clusters records into blocks where they share the identical blocking key. A blocking key is defined to be composed from the record attributes in each data set. Assuming two data sets with n records each are to be linked, and the blocking method resulted in b blocks (all of the same size containing n/b records), the resulting number of record pair comparisons is O(n2 b).

The Sorted Neighbourhood method sorts the records based on a sorting key and then moves a window of fixed size w sequentially over the sorted records. Records within the window are then paired with each other and included in the candidate record pair list. [2]

A. Bigram Indexing

The Bigram Indexing method as implemented in the Febrl record linkage system allows for fuzzy blocking. The basic idea is that the blocking key values are converted into a list of bigrams sub-lists

of all possible permutations will be built using a threshold (between 0.0 and 1.0). The resulting bigram lists are sorted and inserted into an inverted index, which will be used to retrieve the corresponding record numbers in a block. The number of sub-lists created for a blocking key value both depends on the length of the value and the threshold. The lower the threshold the shorter the sub-lists, but also the more sub-lists there will be per blocking key value, resulting in more (smaller blocks) in the inverted index.

B. Canopy Clustering with TFIDF

Canopy Clustering with TFIDF (Term Frequency/Inverse Document Frequency) forms blocks of records based on those records placed in the same canopy cluster. A canopy cluster is formed by choosing a record at random from a candidate set of records and then putting in its cluster all the records within a certain loose threshold distance of it. The record chosen at random and any records within a certain tight threshold distance of it are then removed from the candidate set of records. We use the TFIDF distance metric, where bigrams are used as tokens. . The number of record pair comparisons resulting from canopy clustering is O(fn2 c). where n is the number of records in each of the two data sets, c is the number of canopies and f is the average number of canopies a record belongs to. The threshold parameter should be set so that f is small and c is large, in order to reduce the amount of computation.[2]

V. CONCLUSIONS

This paper presents of six indexing techniques with a variations of them.The number of candidate record pairs generated by these techniques has been estimated theoretically, and their efficiency and scalability has been evaluated using various data sets. These experiments highlight that one of the most important factors for efficient and accurate indexing for record linkage there are large differences in the number of true matched candidate record pairs generated by the different techniques, but also large differences for several indexing techniques depending upon the setting of their parameters. And developed a new technique of indexing.

REFERENCES

[1] Peter Christen ," *A Survey of Indexing Techniques for Scalable Record Linkage and Duplication*" Ieee Transactions On Knowledge And Data Engineering, Vol. Z, No. Y, Zzzz 2011

[2] Rohan Baxter, Peter Christen , Tim Churches," *A Comparison of Fast Blocking Methods for Record linkage"*

[3] Matthew Michelson and Craig A. Knoblock **,"** *Learning Blocking Schemes for Record Linkage*" American Association for Artificial Intelligence.

[4] Mikhail Bilenko, Beena Kamath, Raymond J. Mooney," *Adaptive Blocking: Learning to Scale Up Record Linkage"* To Appear In Proceedings Of The 6th Ieee International Conference On Data Mining, Hong Kong, December 2006.

[5] Uwe Draisbach, Felix Naumann," *A Comparison and Generalization of Blocking and Windowing Algorithms for Duplicate Detectio*. International Workshop on Quality in Databases (QDB) August 24, 2009,Lyon, France.

[6] http://en.wikipedia.org/wiki/Database_index

[7] http://en.wikipedia.org/wiki/Database_index#Types_of_indexes