CHAPTER 7

CONCLUSIONS AND SCOPE FOR FURTHER STUDY

7.1 Conclusions

This research work has reviewed an important data extraction issue occurs during analysis of data set. The basic foundation for any analysis is the execution of aggregate function effectively. Only aggregate functions independently are not capable of analyzing the data set. It follows some structure or syntax like Iceberg query. The main feature of iceberg query is the extraction of the small result set from the large data set. The result of query is small so to extract small data set time required must be less. This is the basic logic behind bitmap preprocessing strategy for Iceberg query evaluation. This is the main property of Iceberg query, utilized in this research to achieve its objective. This research aims to provide an efficient framework for Iceberg query with all aggregate functions like COUNT, SUM, AVG, MIN and MAX. Four types of studies have been done to achieve this objective. These are bitmap preprocessing using LAMT, TPT, PPBT and Cache mechanism.

Many iceberg query processing methods are available as mentioned in chapter 2. But none of the research utilized the property of iceberg query for its evaluation. All of them treat the iceberg query as normal query and due to this time required to evaluate iceberg query using traditional_BI technique is more. Also, none of the previous work provides the complete framework for all aggregate function. This research focuses on addressing the problems related to traditional algorithms like empty bitwise AND, OR and XOR operation as well as fruitless operations. Above problems also incurred futile queue operation problem in traditional_BI technique. Due to all these problems the performance of iceberg query using traditional_BI strategy is reduced.
Bitmap preprocessing strategy for iceberg query evaluation is presented to improve the performance of Iceberg query regarding the time required to evaluate it. The first finding of this research is LAMT, which focus on reducing fruitless bitwise operation problem occurs in the traditional method. It uses the concept of partitioning the vectors into subpart and work on the group of vectors independently. By observing and analysing the intermediate results, it declares final iceberg query result in advance. The second finding of this research is TPT, TPT intends to minimize the empty bitwise operation and reduce the queue maintenance cost. It focuses on bit by bit of logical AND, OR and XOR operation. It continuously keeps track on next bitwise operation result and compare it with threshold limit. In this way in between it declare the query result and skip the further operations.

The third finding of this research is PPBT, it works on the concept of assigning priority to the vectors and then checking the probability of them to be the part of final iceberg query result. In this way intermediate result is compared with threshold limit and query result is declared in advance. It also contributed to avoid fruitless operations and reduces query processing time. The fourth finding of this research is implanting cache mechanism above all these modules. Cache is used here as intermediate storage of iceberg query result. Next time if the same query occur then result will directly fetch from the cache. In this way, it helps to reduce query processing time.

The rigorous experimentation has been performed and recorded. In case of bitmap preprocessing, strategy time saved compare to traditional_BI is up to 81.98% on the 80K dataset, 70.75% on the 40K dataset, 70.01% on the 20K dataset, 50.56% on the 10K dataset and 43.74 % on 5K data set. The effectiveness of bitmap preprocessing strategy is observed through above result. As the data set size increases the performance of bitmap preprocessing also increases.
Iterations required to evaluate iceberg query using bitmap preprocessing strategy is also reduced up to 79.66%. It gets varies from an aggregate function to function. However, it is the major parameter which reduces the time required to evaluate iceberg query. The results related to individual aggregate functions are also noted. Finally, the impact of cache on all the techniques is recorded. Time saved using cache for COUNT aggregate function on all dataset is 12.48%, SUM function it is 6.05%, AVG function it is 2.95%, MAX function 7.68% and for MIN function it is 6.69%.

By experimental results it has been observed that number of iterations required and time required to execute query get reduced to 50 - 57% even though dataset size increases. Experimental results prove the superiority of bitmap preprocessing strategy for iceberg query evaluation by comparing it with traditional_BI method. It minimizes all the problems occurred in previous research and provided an efficient framework for iceberg query evaluation using aggregate functions like COUNT, MIN, SUM, MAX and AVG.

7.2 Scope for Further Study

Further study can be made to improve the performance of Iceberg query in terms of time. The scope is to integrate randomization and predictive algorithm along with Priority and probability based technique of bitmap preprocessing strategy. The probability of randomly selected data sub part can be analysed using predictive method which help to reduce the query response time. Randomization algorithms save the time of assigning priority to data sub part and predictive algorithm help to find the probable sub part of data set which contribute in final Iceberg query result.

This research concentrates only on structured data analysis. Hence, it is applicable for relational databases only. In future, this work can be extended for processing of unstructured data. Unstructured database is a collection of various
types of data. Numerical and character data can be analysed using bitmap preprocessing strategy. The knowledge can be discovered from textual data by introducing some text mining algorithms in bitmap preprocessing strategy. Iceberg query concept along with unstructured dataset will be useful in the field of Information retrieval. It will add some analysis of data to be retrieved. This type of knowledge extraction queries can be integrated into the search engines to extract the precise information effectively and efficiently.

The bitmap preprocessing strategy for Iceberg query has scope in further research in the domain of data processing, query processing and query optimisation. The bitmap preprocessing strategy for Iceberg query has research potential in data analytics system, business Intelligence system, multi join complex queries, unstructured database and knowledge discovery system.