International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

OVERVIEW OF TOP-K QUERY PROCESSING IN RELATIONAL DATABASES

Neha Singh, P.K. Pandey*, Anil Kumar Tiwari**

Department of Computer Science, A.P.S. University, Rewa (M.P.)

******* *Department of Physics, Govt. Science College, Rewa (M.P.)*

*** Department of Physics and Comp. Science, Govt. T.R.S. College, Rewa (M.P.)*

ABSTRACT

Query processing is a fundamental part of Database management system. As the amount of text data stored in relational databases is increasing, it is necessary to support the Top-k query processing over text data. The main objective of top-k query processing is to return the k highest ranked results quickly and efficiently. In this paper, we introduce the Top-k query processing in relational database system. We also discuss the classification of Top-k query processing techniques in relational databases with different design dimensions.

1. Introduction

Emerging application that depends on ranking queries warrant efficient support of ranking in relational database management systems [choudhury et al, 2005]. A relational database consists of a collection of relations, also known as tables each of which is

assigned a unique name. Top-k queries are a long studied topic in the database and IR communities. In many applications, users are interested in the most important query, also known as top-k query answers according to their specified ranking function. A top-k query returns the subsets of most relevant results instead of all results to minimize the cost metric that is associated with the retrieval of all results and maximize the quality of the result set, such that the user is not overwhelmed with irrelevant results [zeinalipour, 2010]. Top-k query processing connects to many database research areas including query optimization, indexing methods and query languages. As a consequence, the impact of efficient top-k processing is becoming evident in an increasing number of applications [Ilyas et al, 2008].

2. Top-k Queries:

The top-k query is define as- Given a database D of m objects, each of which is characterized by n attributes, a scoring function f, according to which we rank the objects in D, and the number of expected results k. Then a top-k query Q returns the k objects with the highest rank in f.

In Top-k query, query define on n attribute a1, a2 ,…, an and relation M in the form of R1 ,R2 ,…,RM that each ai (i=1:n) belongs to one relation Rj (j=1:M). Each of the attributes has special domain in comparison with their kind. According to the query, a series of attributes of these relations are applied for projection, a series of attributes of these relations are used for restriction and join. In the rank aware queries there is apart for ranking that some of relations attribute are presented in the form of a ranking relation which is called ranking function. Ranking function f is formed in the form of attribute n' that is n' <=n. A theory for ranking function f is this: ranking function changes in comparison with all relations are monotonic. In addition to this, the number of suitable

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

answers in rank aware queries is determined too that is just Top k. Consider a set of relations R1 to Rm. Each tuple in Ri is associated with some score that gives it a rank within Ri. The top-k join query joins R1 to Rm and produces the results ranked on a total score. The total score is computed according to some function, say F that combines individual scores. Note that the score attached with each relation can be the value of one attribute or a value computed using a predicate on a subset of its attributes. A possible SQL-like notation for expressing a top-k join query is as follows:

```
SELECT *
FROM R1, R2, . . . ,Rm
WHERE join condition (R1, R2, . . . ,Rm)
ORDER BY F (R1.score, R2.score. . . Rm.score)
LIMIT k;
```

Where LIMIT limits the number of results reported to the user.

3. Top-k Query Processing Techniques:

The classification of top-k query processing techniques based on multiple designs is shown in figure 1. In this section we introduce the various classification dimensions of top-k query processing techniques and their impact on the design of the underlying top-k query processing techniques

3.1 Query Model Dimension

Top-k processing techniques use different query models to specify to score the data objects.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

3.1.1 Top-K selection: Query Model: In this model, the scores are assumed to be attached to base tuples. A top-k selection query is required to report the k tuples with the highest scores. Scores might not be readily available since they could be the outcome of some user-defined scoring function that aggregates information coming from different tuple attributes. Consider a relation R, where each tuple in R has n attributes. Consider m scoring predicates, p1 · · · pm defined on these attributes. Let F(t) = F(p1(t), . . . , pm(t)) be the overall score of tuple t □ R. A top-k selection query selects the k tuples in R with the largest F values.

A SQL –like notation for top-k selection query is the following:

SELECT some attributes
FROM R
WHERE selection condition
ORDER BY F(p1, . . . , pm)
LIMIT k

3.1.2 Top-k Join Query Model: In this model, scores are assumed to be attached to join results rather than base tuples. A top-k join query joins a set of relations based on some arbitrary join condition, assigns scores to join results based on some scoring function, and reports the top-k join results. Consider a set of relations R1 · · Rn. A top-k join query joins R1 · · · Rn, and returns the k join results with the largest combined scores. The combined score of each join result is computed according to some function F(p1, . . . , pm), where p1, . . . , pm are scoring predicates defined over the join results. A possible SQL template for a top-k join query is

SELECT *
FROM R1, . . . , Rn
WHERE join condition(R1, . . . , Rn)

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

ORDER BY F(p1, . . . , pm)
LIMIT k

3.1.3 Top-k Aggregate Query Model.: In this model, scores are computed for tuple groups, rather than individual tuples. A top-k aggregate query reports the k groups with the largest scores. Group scores are computed using a group aggregate function such as sum. Consider a set of grouping attributes G= {g1. . . gr }, and an aggregate function F that is evaluated on each group. A top-k aggregate query returns the k groups, based on G, with the highest F values. A SQL formulation for a top-k aggregate query is
SELECT g1. . . gr , F
FROM R1. . . Rn
WHERE join condition (R1. . . Rn)
GROUP BY g1. . . gr
ORDER BY F
LIMIT k

3.2 Query and Data Uncertainty Dimension

In some query processing environments, for example, decision support or OLAP, obtaining exact query answers efficiently may be overwhelming to the database engine because of the interactive nature of such environments, and the sheer amounts of data they usually handle. The uncertainty in top-k query answers might alternatively arise due to the nature of the underlying data itself. Top-k processing techniques based on query and data certainty is classified as follows:

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

3.2.1 Exact methods over certain data: This category includes the majority of current top-k processing techniques, where deterministic top-k queries are processed over deterministic data.

3.2.2 Approximate methods over certain data: This category includes top-k processing techniques that operate on deterministic data, but report approximate answers in favor of performance.

3.2.3 Uncertain data: This category includes top-k processing techniques that work on probabilistic data. The research proposals in this category formulate top-k queries based on different uncertainty models.

3.3 Data Access Dimension

Many top-k processing techniques involve accessing multiple data sources with different valuations of the underlying data objects. For example each search engine can be seen as a ranked list of web pages based on some score. It is due to the top-k processing techniques. Top-k processing techniques based on the available data access methods in the underlying data sources, is classified as follows:

3.3.1 Both sorted and random access: In this category, top-k processing techniques assume the availability of both sorted and random access in all the underlying data sources.

3.3.2 No random access: In this category, top-k processing techniques assume the underlying sources provide only sorted access to data objects based on their scores.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

3.3.3 Sorted access with controlled random probes: In this category, top-k processing techniques assume the availability of at least one sorted access source. Random accesses are used in a controlled manner to reveal the overall scores of candidate answers.

3.4Implementation Level Dimension

Integrating top-k processing with database systems is addressed in different ways by current techniques. One approach is to embed top-k processing in an outer layer on top of the database engine. Another approach is to modify the core of query engines to recognize the ranking requirements of top-k queries during query planning and execution. This approach has a direct impact on query processing and optimization. Top-k processing techniques based on their level of integration with database engines are classified as follows:

3.4.1 Application level: This category includes top-k processing techniques that work outside the database engine. Some of the techniques in this category rely on the support of specialized top-k indexes or materialized views. However, the main top-k processing remains outside the engine.

3.4.2 Query engine level: This category includes techniques that involve modifications to the query engine to allow for rank-aware processing and optimization. Some of these techniques introduce new query operators to support efficient top-k processing.

3.5 Ranking Function Dimension

The properties of the ranking function largely influence the design of top-k processing techniques. One important property is the ability to upper bound objects' scores. This

property allows early pruning of certain objects without exactly knowing their scores. Top-k processing techniques based on the restrictions they impose on the underlying ranking function is classified as follows:

3.5.1 Monotone ranking function: Most of the current top-k processing techniques assume monotone ranking functions since they fit in many practical scenarios, and have appealing properties allowing for efficient top-k processing.

3.5.2 Generic ranking function: A few recent techniques address top-k queries in the context of constrained function optimization. The ranking function in this case is allowed to take a generic form.

3.5.3 No ranking function: It is also known as Skyline queries. Many techniques have been proposed to answer skyline-related queries. A skyline query returns the objects that are not dominated by any other objects restricted to a set of dimensions

CONCLUSION

In this paper, we introduce the top-k query techniques and its different classification dimensions in relational databases. Top-k Query processing is essential for large information retrieval system. Top-k processing techniques such as the adopted query model, data access, implementation level, and supported ranking functions, are classified based on the restrictions they impose on the underlying ranking or scoring function.

REFERENCES

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 1 Issue 2 July 2011

---

[1] S. Chaudhury, R. Ramakrishnan, and G. Weikum (2005), Integrating db and ir technologies, CIDR.

[2] Ilyas I.F, Beskales, G, Soliman M. A, (2008), A survey of top-k query processing techniques in relational database systems. ACM Comput. Surv. 40, 4, Article 11.

[3] Ilyas I.F, Beskales, G, Soliman M. A, (2008), A survey of top-k query processing techniques in relational database systems. ACM Comput. Surv. 40, 4, Article 11.

[4] Zeinalipour D, (2010), Ranking Query Results in a Networked world. Messaging and Event Systems Department, IBM T.J. Watson Reasearch Center, Hawthorne, NY 10532, USA.