

# Incremental Query Construction and False hits based on Ranking

Ayesha Jabeen  
*M.Tech.(CS)*

Mohd. Anwar Ali  
*Associate Professor*

T. Ram Babu  
*Assistant Professor*

Alhabeeb College of Engineering and Technology. Hyderabad. A.P. India.

## Abstract

*Users find difficulty to retrieve their intended data from the large databases or data warehouses. Users have to create a complex query in order to retrieve the exact information. Otherwise the users have to make many interactions in order to reach the exact result. Incremental Query Construction allows the users to issue keyword queries without any knowledge of the database schema or of SQL. The number of interactions will be reduced with the benefit of ranking.*

*Sometimes users may not find the information with the keywords that the user had entered which results in false hits. In this Paper we store these false hits in a separate table and later provide these to the admin so that he can update the database.*

**Keywords:** Search, Ranking, False hits

## 1.Introduction

In the existing system users may find the irrelevant result while searching the details. To retrieve the precise data from the databases,

users have to create the structured queries which is a very complex process and takes much time. Incremental Query Construction allows the users to create query incrementally

which improves the search efficiently and leads to specific results. Queries can be constructed starting with the keywords and can be proceeded with the required options. These options are listed rank wise to reduce the number of interactions. Users can select the required option corresponding to which the result is displayed. Even the less frequent information can be found with the minimal number of interactions because of incremental query.

This paper consist of three components:

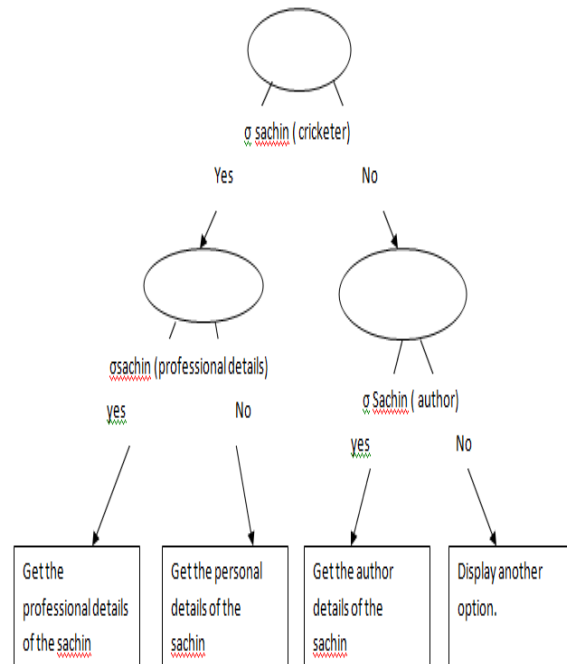
1. Incremental query construction
2. Ranking queries and subqueries
3. Storing and displaying the false hits in ranking order.

Details of each of these components is given in the following sections.

## 2.Overview

The user interface of Incremental Query Construction consists of

1. A search field for users to enter keyword
2. A window to present the query options related to keyword
3. A window to display ranked list of sub queries based on the option selected
4. A result window for presenting search results
5. A window for the admin to check the false hits if any.



**Figure: 1 Binary Tree Implementation**

When a user issues a keyword query, IQP provides the user with a ranked list of structured queries (as interpretations of the keyword query) and the corresponding results, which are presented in the query and result windows, respectively. If the user identifies the desired structured query, they can double click the query to retrieve its results.

## 3.Incremental Query Construction

Incremental Query Construction Plan can be described by using a binary tree or n-ary tree.

The figure shows the example of finding the date of birth of Sachin Tendulkar who is a cricketer.

The logic behind Incremental Query Construction is described in the above figure. If the user enters the keyword “sachin” in the search field of the user interface then the options for the keyword entered are displayed such as sachin as cricketer or sachin as author and so on according to the rankid in descending order. Suppose if the user selects sachin as cricketer then for this query, the subquery options are displayed as personal details of sachin or professional details of sachin. Depending on the user’s selection further options are displayed until he reaches the exact result.

The following procedure explains the coding style of Incremental Query Construction

Proc QP(keyword)

Input: keyword

Output: Result

Logic:

While (true)

Generate query construction options(qop) in which the keyword is found

If qop is null then store the keyword in False Hits

Else display the qop with the rank ids in descending order

If qop is accepted then display all the child options of that qop

Else if qop is rejected then don't display the child options of that qop

Endif

End while;

Finally display the result of the accepted child qop.

Endproc;

#### 4.Ranking

The concept of ranking deals with displaying the information based on the priorities of the users. The query which most of the users search for will be displayed first. Ranking helps the users to find their desired results in top ranks rather than searching for the more number of interpretations.

To implement the ranking concept, one id for each option will be created in the ranking table which will be updated each time the users select that particular option. In this way the options whose id has the highest number will be displayed first that is, the options are displayed in the descending order of their ids. This concept provides much benefit to the users whose results is in the top ranks. But this also helps the users who are searching for least ranked results cause they don't have to waste time

in searching the details of top ranked results, they can simply reject it and search for the appropriate one.

One example for creating the Ranking table is shown below:

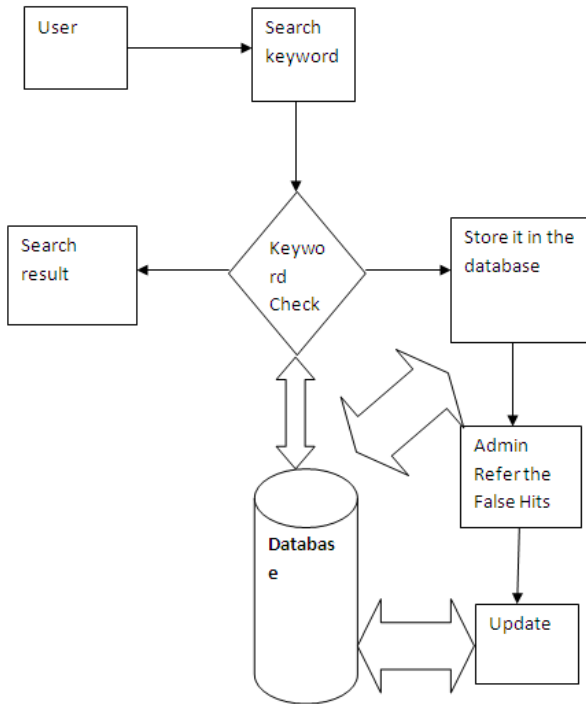
Column Name	Data Type	Allow Nulls
SubcategoryId	int	<input type="checkbox"/>
SubCategory	nchar(10)	<input checked="" type="checkbox"/>
Ranking	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

**Fig 2: SubCategory Ranking table**

Ranking can be applied to the example shown in the figure 1. Most of the users intent is to search for the cricketer by name Sachin Tendulkar rather than an author. Hence Ranking id will be greater for Cricketer than for the author. Similarly the Professional details of Sachin will have the highest rank than the Personal details. This rank id will be updated as and when the users view these details. Thus ranking helps the users to find their intended result quickly.

#### 5.False Hits

In this module we develop the code for storing the False hit Keywords given by the Users. False hit Keyword means the keywords which are not available in the existing database table. Those keywords will be stored in separate table and these keywords will be shown to the Admin page for updating the details.



**Figure 3: Interaction of user and Admin with the database**

The data stored in False hits table will be in ranking order so that it is easy for the admin to update the highest rank information first. Once the admin update the details of False hits keywords in the database, he can delete the information from the False hits table.

## 6. Conclusion

Incremental Query Construction allows the users to identify the meaningful items to interact with the database. With this algorithm users can generate the intended optimal query with the keywords and retrieve the results with the minimal number of interactions. It also helps the Admin to update the database with the missing information.

## 7. Acknowledgement

Our sincere thanks to Mr. MNA Baig for comments and the ideas he proposed.

## 8. References

- [1] Sanjay Agrawal, Surajit Chaudhuri, and Gautam Das. DBXplorer: A System For Keyword-Based Search Over Relational Databases. *ICDE*, 2002.
- [2] Androutsopoulos, G.D. Ritchie, and P. Thanisch, "Natural Language Interfaces to Databases—An Introduction," *J. Language Eng.*, vol. 1, no. 1, pp. 29-81, 1995.
- [3] H. Bast, A. Chitea, F. Suchanek, and I. Weber, "ESTER: Efficient Search on Text, Entities, and Relations," *Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR)*, 2007.