

Ameliorated Methodology For Retrieval Of Data Using Query By Example

Dr. R. N. Kulkarni¹, Ms.Chinmayi A², Ms.Meghana A³, Ms.Sai Harika K⁴,
Ms.Saraswathi Y⁵

¹ Department of Computer Science & Engineering, Ballari Institute of Technology and Management, Ballari, Karnataka, India.

^{2,3,4,5} VIII semester BE(CSE), Ballari Institute of Technology & Management Ballari, India.

Abstract — Query By Example (QBE) is a database query language for relational databases. The Query By Example uses visual tables where the user enters the commands and the output is displayed in a table form.

In this paper, we are proposing an automated tool where the user gives the input in the form of natural language statement (query). The tool processes the input query and retrieves the relevant matching field(s) or record(s) from the input database. The proposed tool comprises a parser where it translates the natural language statement into the tokens and retrieves the appropriate data from the database and displays the output in the form of a table. The tool takes the input from the user in the form of a natural language statement which comprises regular expression. In the proposed tool only the retrieval of data from the database is proposed which works with the print (P) command.

Keywords — Query By Example, Tokens, Parser, query, database.

I. INTRODUCTION

Query by Example (QBE) is a database query language for relational databases. It was devised by Moshé M. Zloof at IBM Research during the mid-1970s, in parallel to the development of SQL. The Query By Example uses visual tables where the user enters the statements and the output is displayed in a table form.

The motivation behind QBE is that a parser can convert the user's actions into statements expressed in a database manipulation language, such as SQL. In Query by example the user doesn't have to specify the structured query explicitly as in SQL. It is easy to learn especially for non-specialists. The user doesn't have to remember the name of the attributes or relations, because they are displayed as part of the template. QBE is especially suited for queries that are not too complex and can be expressed in terms of a few tables.

II. LITERATURE SURVEY

In the paper [1], the article discusses about Visual query systems and languages emerged to alleviate the end-user data access problem by providing intuitive and natural end-user experiences. This article, having Optique as a motivating scenario, is concerned with ontology-based visual query formulation for querying relational databases for end users with no technical skills and knowledge, such as on programming, databases, query languages, and with low/no tolerance, intension, or time to use and learn formal languages as like in SQL and XQuery. This paper helped us to input the data in natural language statement instead of remembering the syntax as in sql.

In the paper [2], the author discusses about how to create queries by designing of template, rather than a language using predicate criteria (as in SQL). This paper helped us to design a template and process the given query.

In the paper [3], the author discusses about searching and parsing the query in a search field. This paper helped us how to conduct a search for data based on matching text, the user enters text into the form search field. This is then passed to the QBE parser for processing. The QBE parser performs an analysis of these and formulates a query to submit to the search engine where the analysis is done in database.

In the paper [4], the author discusses about the differences between SQL and Query By Example. This helped us to differentiate SQL query input statements and QBE input statements.

In the paper [5], the author discusses about the notion of summary table and a high level nonprocedural language, Summary-Table-By-Example (STBE) to manipulate summary data in databases. STBE is similar to Query-by-Example in that it uses graphical two-dimensional objects such as relations and summary tables in formulating a relational database query. This paper helped us to format the data in database to a tabular form.

In the paper [6], the author discusses the overview of retrieval, manipulation, and definition by the Query-by-Example language. One of the unique features of this language is that very few concepts need be learned for one to start using this system. This paper helped us to know the operations that can be performed by Query by example.

In this paper [7], the author discusses about syntactic element search and regular expression pattern search. This paper helped us to know about regular expression pattern search.

In [8], the author discusses about the overview of QBE Language and Fundamentals of database was described. This book helped us to create tables in database and to insert data into database.

In [9], this website gives the basic information about Query by example and how to retrieve the data by giving input in the form of commands.

In [10], the author discusses about the Requirements specification as the process of writing down the user and system requirements in a requirements document. The requirements should be clear, unambiguous, easy to understand, complete and consistent. This book helped us to find the requirements required for our project.

III. TERMINOLOGIES

Query By Example- Query By Example (QBE) is a database query language for relational databases. In this automated methodology, the user enters the input in the form of natural language statement and processes the statement and finally retrieves the data from the database in the form of table.

Structured query language-Is a language used in programming and designed for managing the data present in relational database management system. SQL statements have specific syntax which has to be followed for writing SQL queries. All the trained statements are SQL Queries that retrieves the data according to the user requirements.

Database-It is a structured collection of data where data can be stored and accessed through computer system.

Query-Query is requesting the data from the database containing table or multiple tables. A database query extracts the data from the database and converts it into readable form. Usually it is written in language known as SQL.

Parser-It is a compiler or an interpreter used for translation into another language by breaking the data into smaller parts. It parses through the entire input statement and identifies the intents of that particular statement.

Tokens-It is basic unit of programming language that contains series of bits and circulates around a token ring network.

IV. PROPOSED METHODOLOGY

The proposed methodology is illustrated in the figure 4.1

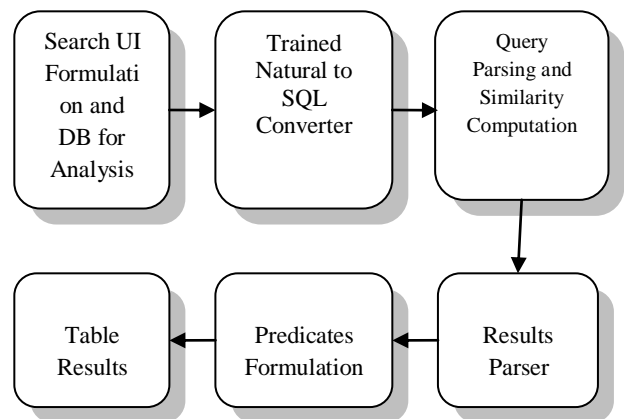


Figure 4.1: Block Diagram of Modules

Search UI Formulation and DB for Analysis

This module is responsible for developing an input page for providing the natural language English statement and sends a request to Query Parser and if no search query is provided then sending back the response.

Trained Natural to SQL Converter

This module is responsible for training the system based on regular expressions. The system will also extract the intents from the query

Query Parsing and Similar Computation

The Query Parsing module is responsible for taking the search query input and converts it into tokenized format.

The details of this module are:

- Divide the statement using a splitter like a SPACE
- The statement will be fed into the First In First Out (FIFO) queue
- Find the number of elements of the queue
- Find the tokens from the trained statements
- Measure the similarity score between the trained statements and query

$$\text{Similarity score} = \frac{\text{Number of elements of query matching trained statements}}{\text{total number of elements}}$$

f) Find the similarity score between each trained statement and query

Trained Statement	Similarity Score
Trained Statement 1	SS1
Trained Statement 2	SS2
Trained Statement n	SSn

Table 4.1: Trained statements with similarity score

g) Find the maximum value of similarity score
 h) The statement which corresponds to maximum similarity score will act like a Trained Statement

Results Parser

The Natural Language is converted into SQL query as per the regular expressions trained statements and intents and then results are returned to the end user.

Table Results

This module is used to return the entire table results with the number of columns on the user interface similar to that of database.

Predicates Formulation

This module is used to represent the predicates for Order By column, Between columns.

The table 4.2 illustrates the list of predicates :

Predicate Label	Meaning	Column Name
Order By	O	Name of Column
SELECT	S	Name of Column
BETWEEN	B	Name of Column
WHERE	W	Name of Column

Table 4.2: List of predicates

V. CASE STUDY

The proposed tool developed which takes the input from the user and displays the required output in the predefined formats.

The following trained statements that need to be used for giving the input by the user. These trained statements are expressed in the form of regular expressions.

1. Need books whose author is ‘*?’
2. Need books based on high price
3. Need books whose cost is between ‘*?’ and ‘*?’
4. Need books whose cost is between ‘*?’ and ‘*?’ and whose author is ‘*?’
5. Need books whose cost is between ‘*?’ and ‘*?’ and whose publisher is ‘*?’
6. Need books based on low price

7. Need books in higher order of price
8. Need books is lower order of price
9. Need books whose publisher is ‘*?’
10. Need books whose cost is between ‘*?’ and ‘*?’ and whose author is ‘*?’ and whose publisher is ‘*?’
11. Need books whose cost is between ‘*?’ and ‘*?’ and whose publisher is ‘*?’ and whose author is ‘*?’
12. I need books whose cost is less than ‘*?’

The natural statement is entered in this query search box, according to given trained statements.

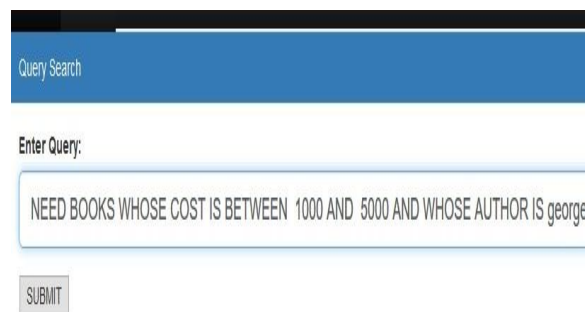


Figure 5.1: Input statement

For example: Consider the following query

Q1. Need books whose cost is between 1000 and 5000 and whose author is George

The intent of the above query is:

- From cost
- To cost
- Author

Now the input is converted into tokenized format

The natural language is converted into sql query as per the regular expression trained statement and final output along with the predicates list is displayed.

Input Statement:

Splitter → need books whose cost is between
 array1 → need books whose cost is between
 array2 → 1000 and 5000 and whose author is George

splitter → and
 array1 → 1000
 array2 → 5000
 array3 → whose author is George

from cost → 1000(array1)

to cost → 5000(array2)

array3 → splitter whose author is

newarray1 → George

author → George

According to these data the query is built and the entire table results are retrieved.

The screenshot shows a 'Book Table Result' with the following data:

Book Name	Publisher/Author	Edition	Book Overview	Book Price	Book Id
Foot Ball World Cup	PLIMTON GEORGE	2nd Edition	In 1960, Plimpton, not an athlete, arranged to pitch to a lineup of professional baseball players in an All-Star exhibition, presumably to answer the question, "How would the average man off of the street fare in an attempt to compete with the stars of professional sports?" He chronicled this experience in his book, Out of My League.	1500.0	3

Below the table is a 'Predicate List' with the following columns and values:

BOOKID	BOOKNAME	CATID	SUBCATID	AUTHOR	PUBLISHER	BOOKPRICE	BOOKOVERVIEW	EDITION	RATING	BOOKPRICE	AUTHOR
SP	SP	SP	SP	SP	SP	SP	SP	SP	SP	BP	MP

Figure 5.2: Output table

VI. CONCLUSION

In this paper, we are proposing an automated tool to retrieve the data from the database using the concept of Query By Example. The proposed tool takes the input given by the user in natural language statement and translates it into tokens, further these tokens are matched with pre-defined statements and generates the predicate list along with the table that contains the data which is required by the user.

VII. REFERENCES

- [1] Ahmet Soylu, Martin Giese, Ernesto Jimenez-Ruiz, Guillermo Vega-Gorgojo, Ian Horrocks, "A Multi-paradigm and Ontology-based Visual Query System for End Users", in Universal Access in the Information Society, 2015.
- [2] Ted Neward, The busy java developers guide to db4o: "QBE", IBM, March 27, 2007.
- [3] Margaret Rouse, "Oracle Definations: Query By Example", Sept. 2005.
- [4] Ramakrishnan, Raghu, Gehrke, Johannes, "Database Management Systems, chap 6: QBE", 3rd ed.
- [5] Z.M. Ozsoyoglu ,G. Ozsoyoglu,"Summary-Table-By-Example: A database query language for manipulating summary data", IEEE First International Conference on Data Engineering, 24-27 April 1984.
- [6] M. M. Zloof,"Query-by-Example: A data base language", IBM Systems Journal, Vol. 16 , No. 4, pp. 324 – 343, 1977.
- [7] VipinBalachandran,"Query by example in large-scale code repositories", IEEE International Conference on Software Maintenance and Evolution (ICSME), 29 Sept-1 Oct 2015.
- [8] Shamkant B.Navathe, Ramez Elmasri, "Fundamentals of Database Systems", 7th ed., Pearson publication, 2016.
- [9] Query By Example –Wikipedia.
- [10] Ian Sommerville, "Software Engineering", 9th ed., Pearson publication,2011.