

# **A Web based Formulation Language for Information Retrieval System in Easy Way**

Voore Subba Rao  
M.Tech.(CSE)  
Asst.Professor, Dept. of CSE  
RG CER, Nagpur(M.S.)

Vishnu Vardhan  
MCA  
Asst.Prof. Dept. of MCA  
RKNEC, Nagpur

Shailendra S.Aote  
M.Tech.(CSE) (Ph.D.)  
Head, Dept. of CSE  
RG CER, Nagpur(M.S.)

## **ABSTRACT**

Formulation is putting together of components in appropriate relationships according to a formula. In databases, Query formulation is an essential part of successful information retrieval. Formulation of valid queries on a relational database, the user must be familiar with SQL known as Structured Query Language. The Information Technology professionals can easily query the data and get information. However, normal users are not familiar with query languages and syntax of a particular query, but would like to access data in a more user friendly way. The goal of this paper is to present a query formulation using a window based application tool that allows users to easily build a query and retrieve the data using an almost natural language, on a step by step basis guided by the tool.

## **Keywords**

Query Formulation, Structured Query Language, Visual query systems, Visual query language, and Graphical user interface.

## **1.INTRODUCTION**

It create a window based Application where is SQL is user friendly. In SQL, customer has to remember the syntax of Query to maintain Database Management which is very difficult. So, to maintain Database in SQL in user friendly format is main motive of paper. The user can use SQL in user friendly format, where customer will be asked every specification of Query in click format. The data will be queried in the form of web. In this paper the software used MS SQL SERVER 2000 for Database and ASP.NET with VB.NET in frontend for Application Development. Being a language is not merely an interface and, at the same time, assuming data to be schema-free is one of the key challenges addressed in the context of design and development of this research paper. Without loss of generality, this article focuses on the Data Web scenario. The Web as a database, where each data source is seen as table.[1] This system under development explore and query one or multiple data sources without prior knowledge about the syntax, IT schema, structure, vocabulary, or any technical details of these sources. This is mainly based on Visual Query Languages represent an evolution, in terms of understandability and adaptability, with respect to traditional textual languages. The iconic query system that enables the interaction of a novice user with a relational database. The paper is to help a user to use and retrieve the relational data from a query language such as SQL. This Visual Query Language is Window based user friendly software applications interacting with the Relational Databases. The system is to provide a language that is easy to use also for non expert users, as it does not require special knowledge of the content of the underlying database or understanding the details of the database schema. The Window based graphical interface, is a visual query processing facility that allows users to query a database through the use of icons. [2]

## **1.1. Advantages**

- Reduction of the time needed to formulate a query by human-computer interaction.
- The graphical click and select representations, where tables are used to represent data and query are performed by table filling.
- The operations performed by user on database essentially use in the form of pointing and clicking operations.
- The queries can be formulated by simple select and click of icons representing the elements (entities and attributes) of the database. The main advantages that this approach presents can be summarized in the following points:.

\* Details about the logical design (tables and attributes) are not provided to the final user. The user can simply identify the select and click objects from which he wants to start to compose its query.

\*The system translates the iconic queries into SQL queries, thus enabling non skilled users to understand and learn the syntax of SQL while composing simple visual queries.

## **1.2 Related work**

The visual representations and direct manipulation interaction mechanisms.

Visual Query Systems (VQSs) may be defined as query systems essentially based on the use of visual representations for that the user can directly interact with the Relational Databases. Window Based Visual Query Systems provide user-friendly query interfaces for accessing a database.

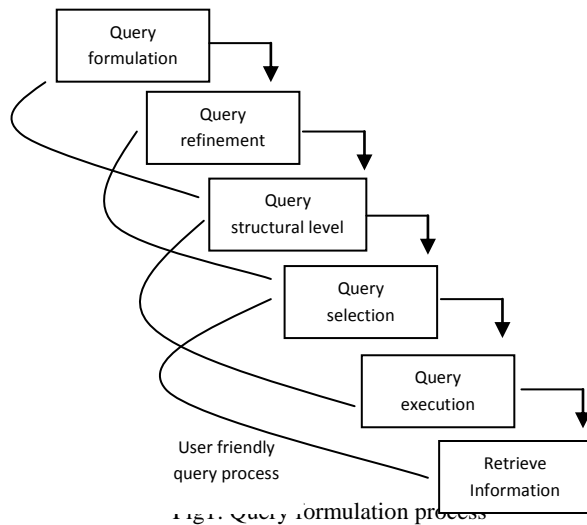
This paper includes both a language to express the queries in a graphical form (i.e., a visual query language, VQL) and a variety of functionalities to facilitate human-computer interaction. The VQSs are oriented to a large variety of users.

## **2. QUERY FORMULATION PROCESS**

The process of query formulation also known as query rewriting or query transformation. In this process modifies the original keyword query submitted by the user to the search engine in order to better represent the underlying intent of the query.

The formulated query is then used as an input to the search engine's ranking algorithm. Thus, the primary goal of query formulation is to improve the overall quality of the ranking presented to the user in response to her query. Query formulation is usually divided into two main processing stages. The first processing stage, which is usually referred to as query refinement [12], alters the query on the morphological level (e.g., tokenization, spelling corrections, stemming, etc.).

After the query refinement stage is completed, the second processing stage alters the query on the structural level for successful information retrieval. In this paper the user retrieve the information in user friendly.



Before formulating a query, one has to know the structure of the data and the attribute labels (i.e., the schema). The users are not expected to investigate “what is the schema”, “how to use schema” and “how to write syntax”. Each time the user search or filter information. In many cases, a data schema might be even dynamic, i.e., many kinds of items with different attributes are often being added and dropped. Other sources might be schema-free, or if it exists.

### 2.1 .Disadvantages of present query language

- Everybody must have a knowledge about the process content would not display if the keyword not matching
- To maintain Database in SQL is not in user friendly format.
- In SQL, the user has to remember the syntax of Query to maintain Database Management which is very difficult
- There are several approaches to solve this problem and hence different solutions exist in the literature about the schema, structure, vocabulary, or any technical details of these sources
- Users not familiar with the Data Web paradigm (i.e., dealing with structured data and the difficulty of querying it).

## 3. PROPOSED SYSTEM

A user friendly visual query formulation language is schema free and in this module, it will create a window based Application SQL make user friendly. The main novelty of this system is that it allows people with limited IT-skills to explore and query one (or multiple) data sources without prior knowledge about the schema, structure, vocabulary, or any technical details of these sources. Being a language not merely an interface and, at the same time, assuming data to be schema-free is one of the key challenges addressed in the context of this Visual system design and development.

### 4.1.2. Describe

Without loss of generality, this article focuses on the Data Web scenario. The Web as a database, where each data source is seen as table.

Customer was given SQL in user friendly format, where customer asked for every specification of Query in click format. The paper placed data onto web, MS SQL SERVER 2000 for Database and ASP.NET with VB.NET in frontend for Application Development.

### 3.1. Advantages

- The main novelty of this system is that it allows people with limited IT-skills to explore and query one (or multiple) data sources without prior knowledge
- Customer will be given SQL in user friendly format, where customer will be asked every specification of Query in click format
- This system is in window based Application which make SQL user friendly
- The user or Customer will be asked every specification of Query in click format
- Query is in user friendly format designed.
- The data to be schema-free is one of the key challenges addressed in the context of this project design and development

## 4. MODULES DESCRIPTION

### 4.1.Data Definition Language (DDL)

CREATE, ALTER,DROP and TRUNCATE are Called DDL Commands. And also known as Data Definition Language used for defining the data. That is the structure of the data is known through these DDL commands. In this module, CREATE, ALTER, DESC, DROP, TRUNCATE commands are in user friendly format. Customer be asked the specification of DDL commands in user friendly format, query will be generated and executed by ASP.NET using VB.NET.

#### 4.1.1.Create

This command used for to Create a table in database. In this paper the create command made web based user friendly command.

```

    Dim x As New globallogic
    x.logic("create database " & TextBox1.Text, "northwind")
    CheckBoxList1.Items.Add(TextBox1.Text)
    End Sub
    If TextBox2.Text <> "" Then
    TextBox5.Text = ""
    TextBox5.Text = "create table " & TextBox2.Text & "("
    End If
    End Sub
    Protected Sub Button4_Click(ByVal sender As Object, ByVal
    As System.EventArgs) Handles Button4.Click
    TextBox5.Text = TextBox5.Text.Substring(0,
    TextBox5.Text.Length - 1)
    DropDownList3.SelectedItem.Value & ""
    End Sub
    End Class
    TextBox5.Text = TextBox5.Text & ")"
    End Sub
    End Class
  
```

This command used for describe the table. In this paper the Describe command made web based user friendly command.

```
Protected Sub DropDownList4_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles DropDownList4.SelectedIndexChanged
    DropDownList3.Items.Clear()
    Dim objcon As New SqlConnection
objcon.ConnectionString = "data source=.; initial catalog=" &
DropDownList4.SelectedItem.Value & "; integrated
security=sspi"
    objcon.Open()
    Dim cmd As New SqlCommand("SELECT * FROM
sysobjects WHERE type = 'U'", objcon)
    Dim dr As SqlDataReader = cmd.ExecuteReader
    While dr.Read
        DropDownList3.Items.Add(dr(0))
    End While
    dr.Close()
    objcon.Close()
End Sub
```

#### 4.1.3 Drop

This command used to Query to remove all objects from database. In this paper the Drop command made web based user friendly command.

```
    DropDownList3.Items.Add("")
loadlist()
End Sub
Protected Sub DropDownList3_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles DropDownList3.SelectedIndexChanged
    TextBox1.Text = ""
    TextBox1.Text = "drop table " &
Protected Sub
DropDownList3_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles
DropDownList3.SelectedIndexChanged
    TextBox1.Text = ""
    TextBox1.Text = "truncate table " &
```

#### 4.2.3 .Update

This command used for update the table. In this technical paper the Update command made easy as web based user friendly command.

```
TextBox2.Text = ""
    TextBox2.Text = "update " &
DropDownList3.SelectedItem.Value & " set " &
DropDownList5.SelectedItem.Value & " = " & TextBox1.Text &
" where " & TextBox3.Text
End Sub
```

#### 4.2.4 Delete

Query for to Delete data in a table. In this technical paper the Delete command made easy as web based user friendly command.

```
objcon.ConnectionString = "data source=.; initial catalog=" &
DropDownList4.SelectedItem.Value & "; integrated
security=sspi"
objcon.Open()
TextBox2.Text = ""
TextBox2.Text = "delete from " &
DropDownList3.SelectedItem.ToString()
Dim objcon As New SqlConnection
```

## 5. CONSTRAINTS

## 4.2.Data Manipulation Language (DML)

DML commands are used for data manipulation, for INSERT, SELECT, UPDATE, DELETE etc. In this module, the paper focus on INSERT/UPDATE/DELETE/SELECT commands in user friendly format. Customer has to ask the specification of DML commands in user friendly format, query will be generated and executed by ASP.NET using VB.NET.

### 4.2.1 Insert

Query for instant DML. In this technical paper the insert command made easy as web based user friendly command.

```
Dim objcon As New SqlConnection
objcon.ConnectionString = "data source=.; initial catalog=" &
DropDownList4.SelectedItem.Value & "; integrated
security=sspi"
objcon.Open()
    Dim thisCommand As New SqlCommand("select * from "
& DropDownList3.SelectedItem.Value, objcon)
    Dim thisReader As SqlDataReader =
thisCommand.ExecuteReader()
Dim schema As DataTable = thisReader.GetSchemaTable()
    Dim i As Integer
    While i < schema.Rows.Count
DropDownList5.Items.Add(schema.Rows(i).Item(0).ToString())
        tp = schema.Rows(i).Item(1).ToString
        i = i + 1
    End While
objcon.Close()
    TextBox2.Text = ""
    TextBox2.Text = "insert into " &
DropDownList3.SelectedItem.ToString() & " values("
End Sub
```

### 4.2.2 Select

This command used for select a table. In this technical paper the select command made easy as web based user friendly command.

```
objcon.ConnectionString = "data source=.; initial catalog=" &
DropDownList4.SelectedItem.Value & "; integrated
security=sspi"
objcon.Open()
Dim thisCommand As New SqlCommand("select * from " &
DropDownList3.SelectedItem.Value, objcon)
objcon.ConnectionString = "data source=.; initial catalog=" &
DropDownList4.SelectedItem.Value & "; integrated
security=sspi"
Dim cmd As New SqlCommand("select * from " &
DropDownList3.SelectedItem.ToString(), objcon)
    Dim sq As New SqlDataAdapter
sq.SelectCommand = cmd
Dim ds As New DataSet
sq.Fill(ds, DropDownList3.SelectedItem.ToString())
    GridView1.DataSource = ds
    GridView1.DataMember =
DropDownList3.SelectedItem.ToString()
    GridView1.DataBind()
End Sub
```

Constraints are used to limit the type of data that can go into a table. Constraints can be specified when a table is created (with the CREATE TABLE statement) or after the table is created (with the ALTER TABLE statement).

In this technical paper Constraints are made easy as web based user friendly commands.

This paper focus on the following constraints are can be used in user friendly way.

- UNIQUE
- PRIMARY KEY
- CHECK

This paper module, makes “UNIQUE, PRIMARY KEY and CHECK” constraints in user friendly format. Customer has to asked the specification of Constraints in user friendly format, query will be generated and executed by ASP.NET using the VB.NET.

### 5.1.Unique Constraint

```
TextBox5.Text = TextBox5.Text & " constraint " &  
TextBox7.Text & "ab unique(" & TextBox7.Text & "),"
```

```
End Sub
```

The Unique constraints uniquely identifies each record in a database table

### 5.2. Primary Key Constraint

The Primary key constraint uniquely identifies each record in a database. In this technical paper the Primary key constraint made easy for web based user friendly command.

```
Protected Sub Button6_Click(ByVal sender As Object, ByVal e  
As System.EventArgs) Handles Button6.Click
```

```
    TextBox5.Text = TextBox5.Text & " constraint " &  
    TextBox7.Text & "ab primary key(" & TextBox7.Text & "),"
```

```
End Sub
```

### 5.3. Check Constraint

The Check constraint limit the value range that can be placed in a column. In this technical paper the Check constraint made easy for web based user friendly command.

```
Protected Sub Button7_Click(ByVal sender As Object, ByVal e  
As System.EventArgs) Handles Button7.Click
```

```
    TextBox5.Text = TextBox5.Text & " constraint " &  
    TextBox8.Text.Substring(0, 2) & "ab check(" & TextBox8.Text &  
    "),"
```

```
End Sub
```

```
End Class
```

## 6. AGGREGATE FUNCTIONS

In this module, the technical paper provide AVG(),COUNT(), MIN(), MAX() and SUM() functions in user friendly format. Customer will be asked the specification of aggregate functions in user friendly format, query will be generated and executed by ASP.NET using VB.NET

The following aggregate functions code written for user friendly usage.

1. Avg()-It returns the average value
2. Count()-The count() returns the number of rows.
- 3.Min()-The min() returns the minimum value.
4. Max()The max() returns the maximum value

## 7. SET OPERATORS

In this paper, technically implement the two types of Set Operators as follows. These set operators are made web based user friendly way.

1.Union

2.Union All

The “UNION(), UNION ALL()” functions in user friendly format. The user will be asked the specification of these Set Operators in user friendly format, query will be generated and executed by ASP.NET using VB.NET

### 7.1 Union

In SQL the UNION clause combines the results of two SQL queries into a single table of all matching rows.

```
TextBox1.Text = ""
```

```
    TextBox1.Text = "select " &
```

```
    DropDownList6.SelectedItem.Value & " from " &
```

```
    DropDownList3.SelectedItem.Value & " union select " &
```

```
    DropDownList7.SelectedItem.Value & " from " &
```

```
    DropDownList5.SelectedItem.Value
```

```
End Sub
```

### 7.2.Union All

In UNION ALL any duplicate records are automatically removed unless UNION ALL is used.

```
TextBox1.Text = ""
```

```
    TextBox1.Text = "select " &
```

```
    DropDownList6.SelectedItem.Value & " from " &
```

```
    DropDownList3.SelectedItem.Value & " union all select " &
```

```
    DropDownList7.SelectedItem.Value & " from " &
```

```
    DropDownList5.SelectedItem.Value
```

```
End Sub
```

## 8.TRANSACTION CONTROL LANGUAGE

A Transactional Control Language (TCL) used to transactional processing in a database. A transaction is logical unit of work that comprises one or more SQL statements. In this research paper the following TCL commands are created in user friendly way.

COMMIT - This command is used to apply the transaction by saving the database changes.

```
TextBox2.Text = ""
```

```
    TextBox2.Text = "insert into " &
```

```
    DropDownList3.SelectedItem.ToString & " values("
```

```
    TextBox3.Text = "commit" &
```

```
    DropDownList3.SelectedItem.ToString()
```

```
    Dim objcon As New SqlConnection
```

```
End Sub
```

*Results:* The performances measured as the ratio of the number of

ROLLBACK - This command is undo all changes of a transaction.

```

TextBox2.Text = ""
TextBox2.Text = "delete from " &
TextBox4.Text="rollback"
DropDownList3.SelectedItem.ToString()
    Dim objcon As New SqlConnection
End Sub
    
```

## 9. EXPERIMENTAL SETTING

An experiment to evaluate in a comparative way the usability of the Visual Query Language Tool for Relational Databases for performance comparing with Oracle SQL and MSSQL. The experiment was conducted in the laboratory. Each student was presented with THREE databases and FOUR groups of queries expressed in natural language, one for each database. The structure of the THREE databases was similar. In particular, each database was associated with FOUR queries. The students were required to formulate the queries for each database with a different query language, namely MSSQL, SQL and user friendly SQL. All queries were executed using in a given PC. The experiment tests are based on check the performance of each query execution and query execution time calculation of minimum time limitation.

The main novelty of this project is that it allows people with limited IT-skills to explore and query one (or multiple) data sources without prior knowledge about the schema and structure of the database.

Future work will be devoted to improve these technical issues. It is worth noting that the iconic approach is a valid support to improve the understanding of SQL, as most student improved their performances on SQL exercises after the use of Visual Query Tool.

The software used are MSSQL SERVER 2000 for Database and ASP.NET in frontend for Application Development.

### 9.1. Results

In performance based on the queries perform by the students using different types of queries using in different types of query languages. The performance calculated and measured as the ratio of the number of resolved queries of each type over the number of students can execute. Tables 1 show the queries perform by the students.

1) Experiment for Performance Testing on execution of queries using different databases for performance testing.

Table1:Applying queries in different Query Languages. Table shows the students get better performance results using Visual SQL Tool.[2]

Database Names	Describe (in %)	Delete (in %)	Rollback (in %)	Average Executed queries (in %)
Oracle SQL	50	65	75	63.33
MSSQL	60	70	80	70.00
Visual SQL Tool	95	85	90	90.00

[7]. M.M. Zloof, "Query-by-Example: A Database Language", IBM

resolved queries of each type and perfect execution of the queries of while using different query languages. In this experiment testing the results shows that while using the Visual SQL tool the students execute more commands execute correctly using user friendly tool

2) Experiment for Performance Testing on minimum time limit executing the query:

In the second experiment conducted for the performance testing, executing the queries using these different RDBMS software.

Table 2: Results of all the queries applying queries in different RDBMS software. The minimum query execution time count in microseconds effects in while using Visual SQL tool [2]

Database Names	Describe query executes (in micro sec 10 <sup>-6</sup> sec)	Delete query executes (in micro sec 10 <sup>-6</sup> sec)	Rollback query executes (in micro sec 10 <sup>-6</sup> sec)
Oracle SQL	875200	825770	975850
MSACCESS	895050	750650	880250
Visual SQL Tool	125500	227612	110205

Results: In above table the students executing the queries while using three different Relational Database Languages Oracle SQL, MSSQL and Visual Query SQL for performance testing measuring for the time limitation of executing queries. The students got very less time limit and performance execution of query while using Visual SQL Tool database for executing queries. The experiment mainly concerned the analysis of the behaviour of users while using different types of database tools. The system i.e. user friendly visual query formulation language for the Data Web is a Window based Application using software VB.NET and ASP.NET.

## 10. ACKNOWLEDGMENTS

Our thanks to the experts who have contributed towards development of the research work.

## 11. REFERENCES

- [1]. Mustafa Jarrar, Marios D. Dikaiakos, "A Query Formulation Language for the Data Web" Journal of IEEE 2011.
- [2]. Lerina Aversano, Gerardo Canfora, Andrea De Lucia, Silvio Stefanucci, "Understanding SQL through Iconic Interfaces" Journal of IEEE.
- [3]. Mustafa Jarrar, Marios D. Dikaiakos "MashQL: A Query-by-Diagram Topping SPARQL -Towards Semantic Data Mashups", ONISW'08, October 30, 2008, Napa Valley, California, USA,1-15.
- [4]. M. Angelaccio, T. Catarci, and G.Santucci, "QBD: A Fully Visual Query System", *Journal on Visual Languages and Computing*, vol. 1,no. 2, 1990, pp 255-273.
- [5]. M. Angelaccio, T. Catarci, and G.Santucci, "QBD: A Fully Visual Query System", *Journal on Visual Languages and Computing*, vol. 1,no. 2, 1990, pp 255-273.
- [6]. M. Angelaccio, T. Catarci, and G.Santucci, "QBD: A Graphical Query Language with Recursion", *IEEE Transactions on Software Engineering*, vol. 16, no.10, 1990, pp 1150-1163.

*Systems Journal*, vol. 16, no. 4, 1977, pp. 324-343.

- [8]. P. Reisner, “Query Languages”, *M. Helander ed. Handbook of Human-Computer Interaction*, Elsevier Science Publ., 1988, pp 257-280.
- [9]. P. Reisner, “Query Languages”, *M. Helander ed. Handbook of Human-Computer Interaction*, Elsevier Science Publ., 1988, pp 257-280.
- [10]. G. Ozsoyoglu, V. Mates, and Z.M. Ozsoyoglu, “Query Processing Techniques in the Summary-Table-by- example Database Query Language”, *ACM TODS*, vol. 14, no. 4, Dec. 1989, pp. 526-573
- [11]. T. Catarci, M.F. Costabile, S. Levialdi, C. Batini, “Visual Query Systems for Databases: A Survey”, Technical Report SI/RR-95/17 of Dipartimento di Scienze dell’Informazione, University of Rome “La Sapienza”, 1995.
- [12]. T. Catarci, M.F. Costabile, S. Levialdi, and C. Batini, “Query Systems for Databases: A Survey”, Technical Report SI/RR-95/17, Università degli Studi di Roma, Sapienza, Dipartimento di Scienze dell’Informazione, 1995.

\*\*\*