

A User Friendly Window Based Application for Calculation of Query Execution Time for Relational Databases

Voore Subba Rao, Vinay Chavan

Abstract - In order to managing and calculating query execution time for Relational Database Management System (RDBMS), one must be fluent in Structured Query Language (SQL). The important concept considered in SQL are (entities, relationships, attributes) and the data schema while using SQL. The user has to remember the syntax of Query to maintain database management which is very difficult. However, normal users are not familiar with query languages and database structures, but would like to know the execution time of queries of various RDBMS languages and access data in a more user friendly way.

Keywords: Execution time, Query execution, User friendly query, Time estimation, Window based application.

I. INTRODUCTION

Most of the relational database application programs are written in high-level languages integrating a relational language. The relational languages offer generally a declarative interface (or declarative language like SQL) to access the data stored in a database. These steps are involved in query processing: decomposition, optimization and execution. The first step decomposes a relational query using logical schema into an algebraic query. In this step syntactic, semantic and authorization are done. The second step is responsible for generating an efficient execution plan for the given SQL query from the considered search space. The third step consists in implementing the efficient execution plan or operator tree. This paper focus on calculating the query execution time for various relational databases like ORACLE, MSSQL SERVER2000 and MS-ACCESS.

In this module, the software create a window based Application where the user will calculate the query execution time user friendly. The main motive is to calculate execution time of the different RDBMS queries in user friendly format. 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, structure, vocabulary, or any technical details of these sources. [2]

The aim of this tool is to present a query formulation using a Window based application tool that allows casual 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.

Techniques are described for managing query execution by estimating and monitoring query execution time. A query to be executed and calculate an initial estimated execution time for the received query.[2]

The aim of the project is to present a query formulation using a Window based application tool that allows casual 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.

In this paper the software are ORACLE, MSSQL SERVER 2000 and MSACCESS for Databases and VB.NET used Front-end for Application Development.

Estimating and monitoring query processing time

The present work related to database to managing query execution by estimating and monitoring query processing time.

Regardless of the particular architecture of RDBMS, a requesting entity (eg. an application or the operating system) in a DBMS requests access to a specified database by issuing a database access request. such may include, for instance, simple catalog lookup requests or transactions and combination of transactions that operate to read, change and specified records in the database. These requests (i.e. queries) are often made using high-level languages such as the structured query language (SQL). Upon receiving such a results, the DBMS may execute the request against a corresponding database, and return any result of the execution to the requesting entity. As databases grow in size and in workload, particular queries may take a substantial amount of time and resources to execute. The system or computer program i.e.VB.NET code is a product for managing the execution of a query. This type of method and the code include receiving a query to be executed. This software method and computer program include the processing of calculating an initial execution time for the received query.

A user friendly calculation of query processing of various databases is schema free 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.

Related work

Visual Query Systems (VQSs) is defined as query systems essentially based on the use of visual representations and the user can directly interact with the Relational

Manuscript published on 30 December 2012.

* Correspondence Author (s)

Voore Subba Rao, Kavikulguru Institute of Technology & Science, Ramtek Dist. Nagpur Maharashtra State.

Dr. Vinay Chavan, Professor & Head Porwal College, Kamptee Nagpur Dist. Maharashtra State

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

A User Friendly Window Based Application for Calculation of Query Execution Time for Relational Databases

This Visual Query Language is Window based user friendly software applications interacting with the Relational Databases. The Relational Database query languages are not very simple to use for users non skilled on database technologies as a consequence of the fact that the interaction is based on a textual language such as SQL.

The code also include upon determining the calculated initial execution time and estimating the execution of query start-time and end-time of the query execution time.[1]

II. SYSTEM ARCHITECTURE3

The system is designed in a Web based application that provides the user with a Graphical User Interface (GUI) exploiting the icon metaphor to query for calculation of query execution of various database. The architecture of the this system is composed of three parts.

- Database component containing the database to be queried and the meta-information necessary to characterize it, in particular tables, attributes and previously saved queries,
- Client component, implemented as a set of applets,
- Server component, that receives data from the client and send it to the database.[3]

This include 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.

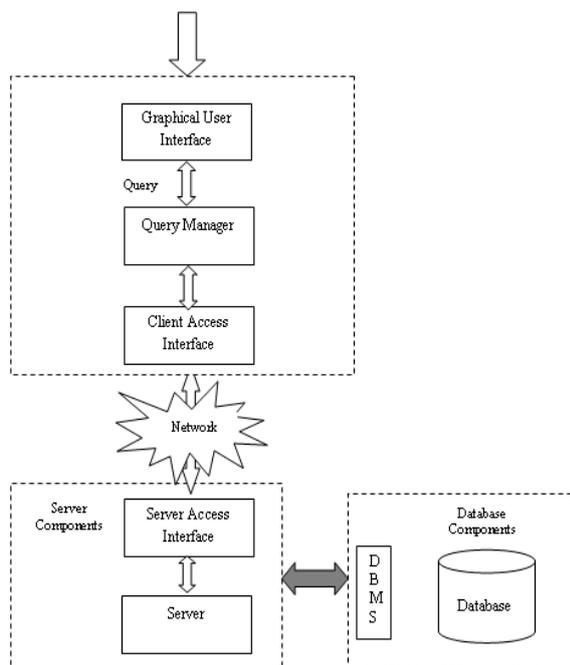


Fig1: System Architecture

III. EXISTING SYSTEM

Before formulating a query, one has to know the structure of the data and the attribute labels (i.e., the schema). End-users are not expected to investigate “what is the schema” each time they 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

Databases for querying while using the user friendly Window Based Visual Query Systems. This provides user-friendly query interfaces for accessing a database. It includes materializing query result, distributing queries, publishing, and discovery of queries, among other issues. 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]

IV. PROPOSED SYSTEM

A user friendly calculation of query processing of various databases is schema free in this module. It will creates 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.

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 our main motive of project.

In this paper SQL in user friendly format, where customer asked for every specification of Query in select and click format. We have placed data onto web, MS SQL SERVER 2000 for Database and ASP.NET with VB.NET in frontend for Application development.

Experiment for Performance Testing on Calculating of query processing time in various databases

The experiment conducted for the performance testing, executing the queries for calculating the query processing time in various RDBMS databases. The code using for calculate the query. The software used are MS SQL SERVER 2000 for Database and VB.NET frontend and code for Application Development.

Flow chart for showing processing steps for calculation of query processing time

The following flow chart is for Calculation of Query processing start and end time results display

```

Dim sqlst As String = ListBox2.SelectedItem
If sqlst = "Select" Then
stime = Now.Ticks
Dim sqlcon As SqlConnection = New
SqlConnection("data source=.;initial
catalog=sqlldb;integrated security=true")
Dim sqlcom As New SqlCommand("select *
from myemp where empno=106", sqlcon)
sqlcon.Open()
Dim sqlldr As SqlDataReader =
sqlcom.ExecuteReader()
If sqlldr.Read() Then
found = True
Else
found = False
End If
sqlcon.Close()
etime = Now.Ticks
edtime = Now
    
```

might be schema-free, or if it exists.

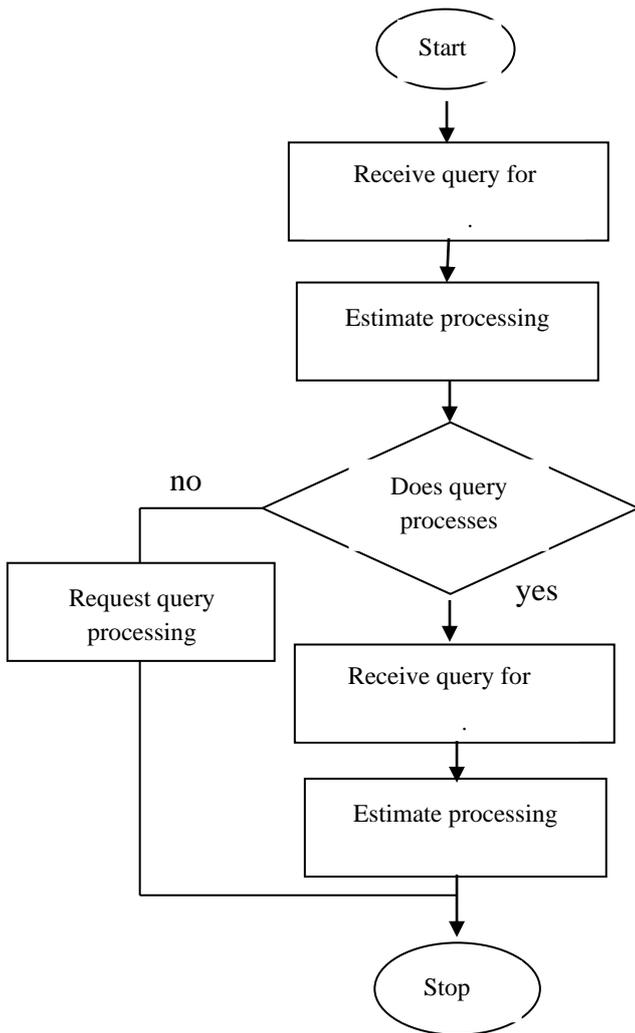


Fig2: Calculation of Query processing start and end time results display

Experiment 1

Code for Performance Testing on SELECT command for Calculating of query processing time in ORACLE, MSSQL2000 and MS-ACCESS.

Code

```

Dim stime, etime As Long
Dim sdtme, edtime As Date
Dim found As Boolean
Private Sub Button1_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Button1.Click
    Dim dbproduct As String = ListBox1.SelectedItem
    If dbproduct = "SqlServer" Then
        sdtme = Now
        MsgBox(stime)
        'Data Source=.;Initial Catalog=sqlldb;Integrated
        Security=True
    
```

The following spanshots are the results displayed using the VB.NET for application development. These results displayed the calculation of query execution time for SELECT command for ORACLE, MSSQL SERVER 2000 and MSACCESS.

```

MsgBox(etime)
Label4.Text = (((etime - stime) / 10) / 1000)
Label6.Text = (((etime - stime) / 10))
Dim sqlcon As OdbcConnection = New
OdbcConnection("dsn=ora;uid=scott;pwd=tiger")
'("Provider=Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=c:\accessdb.acddb")
Dim sqlcom As New OdbcCommand("select *
from myemp where empno=106", sqlcon)
sqlcon.Open()
Dim sqlldr As OdbcDataReader =
sqlcom.ExecuteReader()
If sqlldr.Read() Then
    found = True
Else
    found = False
End If
sqlcon.Close()
etime = Now.Ticks
edtime = Now
'MsgBox(etime)
Label4.Text = (((etime - stime) / 10) / 1000)
Label6.Text = (((etime - stime) / 10))
' Label7.Text = (((etime - stime / 10) / 1000) /
1000) & " seconds"

```

```

Dim sqlcon1 As OleDbConnection = New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0
;Data Source=c:\accessdb.acddb")
Dim sqlcom1 As New OleDbCommand("insert
into myemp values('kiran',106,6000)", sqlcon1)
sqlcon1.Open()
Dim n As Integer = sqlcom1.ExecuteNonQuery()
sqlcon1.Close()
etime = Now.Ticks
edtime = Now
'MsgBox(etime)
Label4.Text = (((etime - stime) / 10) / 1000)
Label6.Text = (((etime - stime) / 10))
'Label7.Text = (((etime - stime / 10) / 1000) /
1000) & " seconds"
Dim sqlcon1 As OdbcConnection = New
OdbcConnection("dsn=ora;uid=scott;pwd=tiger")
Dim sqlcom1 As New OdbcCommand("insert
into myemp values('kiran',106,6000)", sqlcon1)
sqlcon1.Open()
Dim n As Integer = sqlcom1.ExecuteNonQuery()
sqlcon1.Close()
etime = Now.Ticks
edtime = Now
'MsgBox(etime)
Label4.Text = (((etime - stime) / 10) / 1000)
Label6.Text = (((etime - stime) / 10))
'Label7.Text = (((etime - stime / 10) / 1000) /
1000) & " seconds"

```

The following spanshots are the results displayed using the VB.NET for application development. These results displayed the calculation of query execution time for INSERT command for ORACLE, MSSQL SERVER 2000 and MSACCESS.

A User Friendly Window Based Application for Calculation of Query Execution Time for Relational Databases

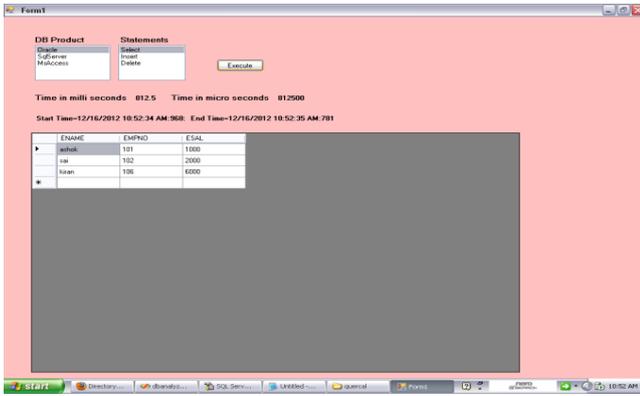


Fig3: Using ORACLE database querying the SELECT command for executing the calculating the query executing time.

Comparative analysis of execution time of SELECT query
 Table 1: Calculation of SELECT query execution time by using ORACLE, MSSQL SERVER and MSACCESS relational databases.

Data-bases	Start time	End time	Milli-seconds (10 ⁻³ sec)	Micro seconds (10 ⁻⁶ sec)
Oracle	12.16.2012 10:52:34 AM:968	12.16.2012 10:52:35 AM:781	812.5	812500
SQL SERVER	12.16.2012 10:55:28 AM:437	12.16.2012 10:55:29 AM:812	1375	137500
Ms-Access	12.16.2012 10:59:03 AM:406	12.16.2012 10:59:07 AM:812	4406.25	4405250

Experiment 2

Code for Performance Testing on INSERT command for Calculating of query processing time in ORACLE, MSSQL2000 and MS-ACCESS.

Code

```
Dim sqlcon1 As SqlConnection = New SqlConnection("data source=.;initial catalog=sqlldb;integrated security=true")
Dim sqlcom As New SqlCommand("insert into myemp values('kiran',106,6000)", sqlcon1)
sqlcon1.Open()
Dim n As Integer = sqlcom.ExecuteNonQuery()
edtime = Now
```

Comparative analysis of execution time of INSERT query
 Table 1: Calculation of INSERT query execution time by using ORACLE, MSSQL SERVER and MSACCESS relational databases.

Databases	Start time	End time	Milli seconds (10 ⁻³ sec)	Micro seconds (10 ⁻⁶ sec)
Oracle	12.16.2012 10:53:24 AM:953	12.16.2012 10:53:25 AM:796	843.75	843750
SQL SERVER	16.2012 10:56:52 AM:375	12.16.2012 10:56:52 AM:609	234.375	234375
Ms-Access	12.16.2012 11:00:13 AM:234	12.16.2012 11:00:13 AM:375	140.625	140625

Fig4: Using MSSQL database querying the INSERT command for executing the calculating the query executing time.

Code

```
Label6.Text = (((etime - stime) / 10))
Label7.Text = (((etime - stime) / 10) / 1000) / 1000 & " seconds"
Label7.Text = "Start Time=" & sdttime.ToString() & ":" & sdttime.Millisecond & ":" & " End Time=" & edtime.ToString() & ":" & edtime.Millisecond & " '(((etime - stime) / 10) / 1000) / 1000) & " seconds"
End Sub
LoadOdbcRecords(sqlcon1)
sqlcon1 = Nothing
System.GC.Collect()
End If
End If
```

Fig5: Using MSACCESS database querying the DELETE command for executing the calculating the query executing time Comparative analysis of execution time of DELETE query

Table 1: Calculation of DELETE query execution time by using ORACLE, MSSQL SERVER and MSACCESS relational databases.

Databases	Start time	End time	Milli seconds (10 ⁻³ sec)	Micro seconds (10 ⁻⁶ sec)
Oracle	12.16.2012 10:54:18 AM:500	12.16.2012 10:54:19 AM:359	859.375	859375
SQL SERVER	12.16.2012 10:57:58 AM:171	12.16.2012 10:57:58 AM:250	78.125	78125
Ms-Access	12.16.2012 11:00:58 AM:859	12.16.2012 11:00:58 AM:921	62.5	62500

V. CONCLUSION

The experiment mainly concerned analysis of the database while conducting experiment for calculating the query execution time by using user friendly window based tool. This window based application allows users to easily query and calculate the query execution time and also retrieve the data and result.

The user can select the particular database and execute the query as per user choice. The user utilizing this user friendly Window based Application for calculating the query execution time.

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.

Experiment 3

Code for Performance Testing on DELETE command for Calculating of query processing time in ORACLE, MSSQL2000 and MS-ACCESS.

Code

```
Dim sqlcom As New SqlCommand("Delete from myemp
WHERE empno=106", sqlcon1)
Dim sqlcon1 As OleDbConnection = New
OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Dat
a Source=c:\accessdb.accdb")
Dim sqlcom1 As New OleDbCommand("Delete from
myemp where empno=106", sqlcon1)
sqlcon1.Open()
Dim n As Integer = sqlcom1.ExecuteNonQuery()
sqlcon1.Close()
etime = Now.Ticks
edtime = Now
'MsgBox(etime)
Label4.Text = (((etime - stime) / 10) / 1000)
Label6.Text = (((etime - stime) / 10))
'Label7.Text = (((etime - stime) / 1000) / 1000) &
" seconds"
Dim sqlcon1 As OleDbConnection = New
OleDbConnection("dsn=ora;uid=scott;pwd=tiger")
Dim sqlcom1 As New OleDbCommand("Delete from
myemp where empno=106", sqlcon1)
sqlcon1.Open()
Dim n As Integer = sqlcom1.ExecuteNonQuery()
sqlcon1.Close()
etime = Now.Ticks
edtime = Now
'MsgBox(etime)
Label4.Text = (((etime - stime) / 10) / 1000)
```

The performances measured with the number of resolved queries of each type and the time calculation of query perfect execution of calculation of execution time SELECT, INSERT, DELETE queries of while using different query languages like ORACLE, MSSQL and MSACCESS. can be used in Semantic Web Pipes. This framework extends ORACLE and MSSQL and allows caching remote sources for querying the database.

Above table shows executing the SELECT, INSERT, DELTE queries while using three different Relational Database Languages ORACLE, MSSQL and MSACCESS for calculating the query execution time. The execution time vary time to time.

The work we have describes a visual query language based on a Window Based Application Select & Click as per user choice. The user perceives the database.

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
- Customer asks as per specification of query in user friendly format
- The data to be schema-free is one of the key challenges

- This user friendly window based calculating query execution time for various databases can be used on LAN using Client Server Technology. Through the LAN the queries are executed and user can estimate the query execution time. The scope of the paper is for optimization of queries for minimum execution of queries. The user analyze the query execution time and utilizing this user friendly Window based Application. It addressed in the context of this project design and development
- Reduction of the time needed to formulate a query by human-computer interaction.
- On improved use of resources, due to the fact that the process of generating and selecting a query based on the conceptual description of the database does not involve the DBMS until the execution of the query.
- 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 icons representing the databases. 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.

REFERENCES

- [1]. Eric L.Barsness, Pine Islad, MN, John M.Santosuess, Rochester "Estimating and Monitoring query Processing Time" IEEE 2012.
- [2]. Mustafa Jarrar, Marios D. Dikaiakos, "A Query Formulation Language for the Data Web" Journal of IEEE 2011.
- [3]. Eric L.Barsness, Pine Islad, MN, John M.Santosuess, Rochester "Estimating and Monitoring query Processing Time" IEEE 2012.
- [4]. Mustafa Jarrar, Marios D. Dikaiakos, "A Query Formulation Language for the Data Web" Journal of IEEE 2011.
- [5]. Lerina Aversano, Gerardo Canfora, Andrea De Lucia, Silvio Stefanucci, "Understanding SQL through Iconic Interfaces" Journal of IEEE.
- [6]. 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.
- [7]. 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.
- [8]. 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.
- [9]. 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.
- [10]. M.M. Zloof, "Query-by-Example: A Database Language", *IBM Systems Journal*, vol. 16, no. 4, 1977, pp. 324-343.
- [9] P. Reisner, "Query Languages", *M. Helander ed. Handbook of Human-*

A User Friendly Window Based Application for Calculation of Query Execution Time for Relational Databases

- Computer Interaction*, Elsevier Science Publ.,1988, pp 257-280.
- [10] P. Reisner, "Query Languages", *M. Helander ed. Handbook of Human-Computer Interaction*, Elsevier Science Publ., 1988, pp 257-280.
- [11] 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
- [12] 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.