

Web Based Application Common Interface for Multiple Databases

¹Jayesh Badhane , ²Kalyani Kanade , ³Pooja Bhor, ⁴Mayur Amrutkar

^{1,2,3,4}Department Of Computer Engineering, Late G.N Sapkal College of Engineering, Nashik, Maharashtra, India.

¹jayeshbadhan97@gmail.com , ²kalyani.kanade17@gmail.com, ³pujabho4471@gmail.com, ⁴mayuramrutkar5@gmail.com.

Abstract - Modern enterprises often deploy multiple Databases form different vendors. Managing a heterogeneous mix of databases is a very challenging exercise to provide solution for this complex administrative task, major database vendors have provided many automatic tool. The tools help automatic common management task and even help in performance tuning. However, DBA now has to phase variety of different databases for different task, each with different interfaces and capabilities. The Applications also suffer from the similar problems when trying to use different databases that help him develop application from Databases. Clearly, there is need for a common interface to manage heterogeneous databases and develop enterprise class Application for them. To address this need we argue for web-based approach to developing a database tool. In particular , we take the case of an index adviser and shows that a web-based approach can provide a common interface for the DBA or the application developer to use across all the database deployment , provide flexibility in the number of an kind of scenarios it can be used , an finally overcome the tool cost of ownership for enterprise .

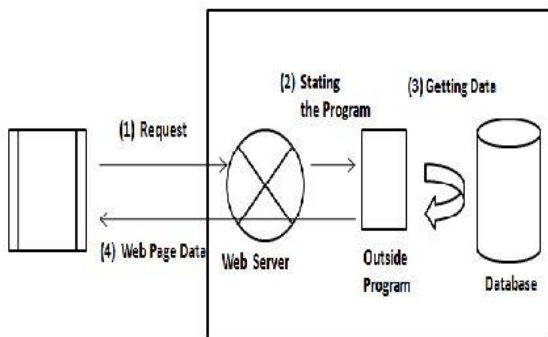
Keywords — Relational Database Management System, Structured Query Language, Open Database Connectivity, Online Transaction Processing.

I. INTRODUCTION

The large amount of data is stored safely with the help of the databases. When database is reliable if it has the capability of adding, Modifying and retrieving data conveniently. In that the retrieving or accessing of data is very important. A collection of programs that facilitate you to store, modify, and excerpt information from a database is called as Database Management System (DBMS). The structured query language (SQL) is one of mostly useful languages which is used for interacting with the database. All the organizations storing the data by using different types of databases because of the advantages of each and

every version of the databases adding extra features or the improvement in the release period of the databases. So if the end-user of the system wants to access the data from these different types of databases then it is challenging for the end-user has to facing the different types of front end applications and to the different types of database structures. So the query builder is invented to access the data from many different types of database such as MS-Access, SQL server, Oracle as that of SQL. The main difference in this is, if we using SQL, data can be accessed only form the Oracle database but when we using the Query Builder data from database like a MS-Access, SQL server, Oracle This query builder is also act as a graphical user interface for the end- user and contains various choices which make the users work very simple or easily. The

QUERY Structure is one of the most extensively used user interfaces for querying databases. The long-established query forms are designed and predefined by developers or DBA in various information management systems. With the fast development of web information and scientific databases, modern databases become very large and complicated. Therefore, it is difficult to design a set of static query forms to satisfy various ad-hoc database queries on those complex databases. Many existing database management and development tools, such as Easy Query. The Data is stored safely with the help of the databases. A database is said to be reliable when if it has the capability of adding collection of programs that facilitate you to store, modify, and extract information from a database is called as Database Management System (DBMS). The structured query language (SQL) is one of the popular language for using to interact with the database.



A. Existing system:

The Query Builder is an application which provides a graphical user interface (GUI) for working with queries in various or different types of database. Complicated queries are created or produce in simple manner against any back-end database engine. It has the integrated data transfer features along which the data may be transferred between various types of data sources. It is such as a part of the RDBMS system because it acts as a tool to access the data from the database. It also has all the advantages or improvements of the RDBMS data access tool. This Query Builder is a GUI application for accessing the data from

many different types of Databases like MS-Access, SQL server, Oracle. It also act like as a Natural Language Interface for Structured Databases. With this application the user can fetch the data from various databases by using a single query. Here the Data Manipulation language it can be used for all the statements of SQL's. Since this application is a Graphical User Interface (GUI), so it is a more user friendly. In addition, in this it also consist of features like printing of result and build new table with existing result.

B. Proposed system:

In this part a connection string is built for the databases, which is a simple process to the user. The connection is formed to the database by using the components such as ADO, OLEDB and ODBC driver. This is made available to the user by means of a simple click. After the connection is formed the list of tables is available in the particular database is displayed in the explorer window. An explorer window is a simple window which is used for display the directory of files of a particular database. Now the relation or a pipeline is establish between the database and the application and so we can collaborate with the structured query language with the database. The query is executed by the end-user with the help of simple actions. Here around all the queries of the data manipulation language such as select, update, edit, delete can be used. The query can be collect by the user by doing various click events without typing the query, and the query can be displayed to the end-user. Here the executed query results can be printed, design as a new table and viewed in maximum size window.

II. LITERATURE SURVEY

In this query builder a single database is connected to this particular application and by using the application the user retrieves data from it by using a query. ACeDB is a database of genome mapping information for the nematode worm *Caenorhabditis elegans*. It is gain through an object-oriented graphical user interface called as Angis query builder. ACeDB is a large program. However, the graphical

nature of the program makes it easy to explore and experiment with. ACeDB it allows the retrieval of these data at various levels, from whole chromosomes down to individual genes.

In the Previous DACS scheme, virtual use methods are used for common gateway interface. This is perform in the case of inventing the DACS Scheme such as a network management scheme, which is a kind of PBNM that we have been proposed before. The DACS Scheme is this scheme which has a control mechanism on each client. HyperSQL can be used to create forms and hypertext-based database interfaces for non-computer experts (e.g., scientists, business users). Such interfaces permit the user to query databases by filling out query forms selected from menus. No knowledge of SQL is required because the interface automatically composes SQL from user input. The Query Builder service is greatly simplifies query formulation, specifically for users unfamiliar with the organization of the database. In query builder, search commands can be chosen from menus rather than typed in. if the query builder window is opened then a query option is available.

III. SYSTEM ARCHITECTURE

In this, we describe the System architecture.

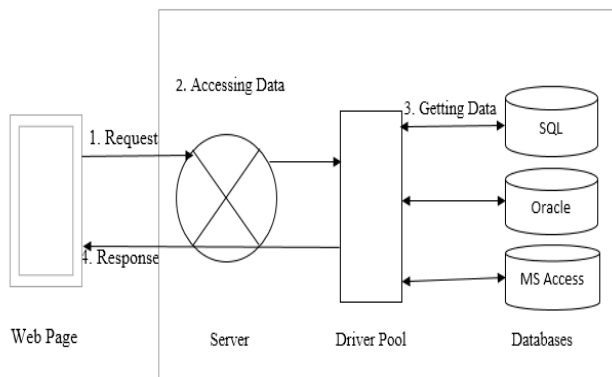


Fig.1. System Architecture

Large amount of data are stored safely with the help of the databases. A database is said to be reliable if it has the

capability of adding, editing and retrieving data efficiently. Here the retrieving or accessing of data is very important. A collection of programs that enables you to store, modify, and extract information from a database is called as Database Management System (DBMS). The structured query language (SQL) is one of the famous languages which is used to interact with the database.

All the organizations store data by using different type of databases because of the advantages of each and every version of the databases and the variation in the release period of the databases. So if the user of the organization wants to access the data from these different types of databases it is difficult because the user has to switch over to the different front end applications or to the different database environments. So the query builder is created to access data from different types of database such as MSAccess, SQL server, Oracle as that of SQL. The basic difference is by using SQL, data can be accessed only form the Oracle database but by using the Query Builder data from database such as MS-Access, SQL server, Oracle This query builder will also act as a GUI for the user and contains various options which make the users work very simple.

IV. RELEVANT MATHEMATICS ASSOCIATED WITH THE PROJECT

A. System Description:

a. Input

- 1) For registration inputs:
 1. Username.
 2. Password.
 3. E-mail id.
 4. Address.
- 2) For login inputs:
 1. Username.
 2. Password.
- 3) For DBconnect:
 1. Username
 2. Host
 3. Database type

b. Output:

Desired Database access.

B. Divide and conquer strategies

In computer science, divide and conquer (D&C) is an algorithm design paradigm based on multi-branched recursion. A divide and conquer algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type (**divide**), until these become simple enough to be solved directly (**conquer**). The solutions.

C. Functional relations

Function overloading or method overloading is the ability to create multiple methods of the same name with different implementations. Calls to an overloaded function will run a specific implementation of that function appropriate to the context of the call, allowing one function call to perform different tasks depending on context. For example, doTask() and doTask(object O) are overloaded methods. To call the latter, an object must be passed as a parameter, whereas the former does not require a parameter, and is called with an empty parameter field. A common error would be to assign a default value to the object in the second method, which would result in an ambiguous call error, as the compiler wouldn't know which of the two methods to use.

S={s, e, X, Y, Fme, DD, NDD, Ffriend, MEMshared, CPUcorecnt}

Here,

s->User will either do registration or login.

e-> Desired Database will be seen.

X-> 1) For registration inputs:-

1. Username.
2. Password.
3. E-mail id.
4. Address.

2) For login inputs:-

1. Username.
2. Password.

3) For DBconnect:

1. Username
2. Host
3. Database type

Y-> 1) Registration Successful or Registration Failed.

2) Login Successful or Failed.

Fme-> load balancing

DD-> stagein <- Fetching Time (cl,v,in)
 Stage out<- Execution Time (cl,v,out)
 Exec<-Execute Time (cl,v)

NDD-> index-> random()
 update(v)

Ffriend-> Logger
 CPUcorecnt -> 01

A. Success Conditions:

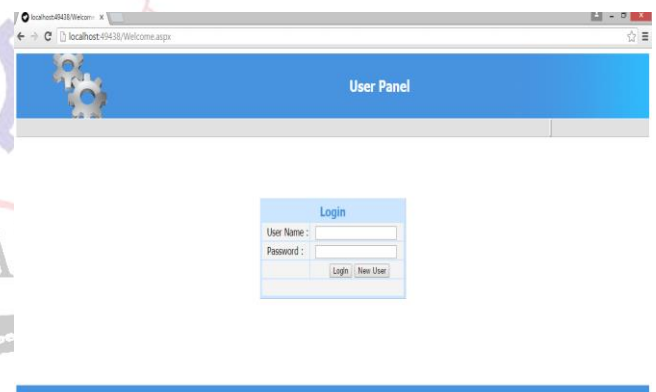
Database accessed successfully.

B. Failure Conditions:

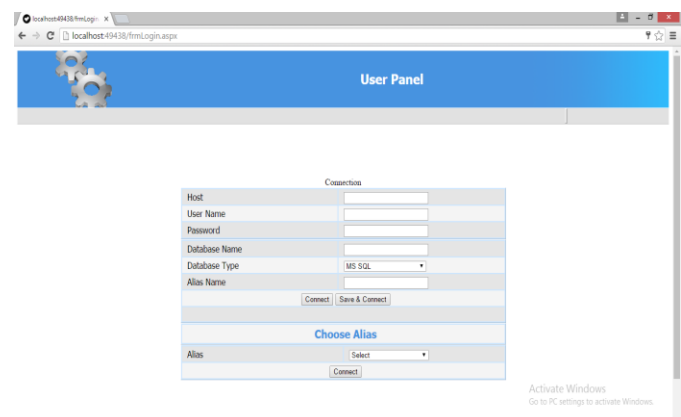
Connection Failed.

V. RESULT ANALYSIS

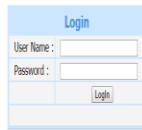
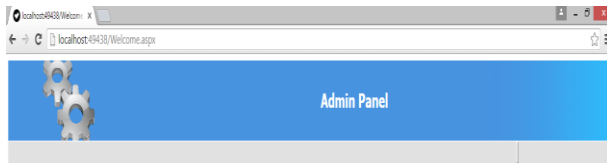
User Login



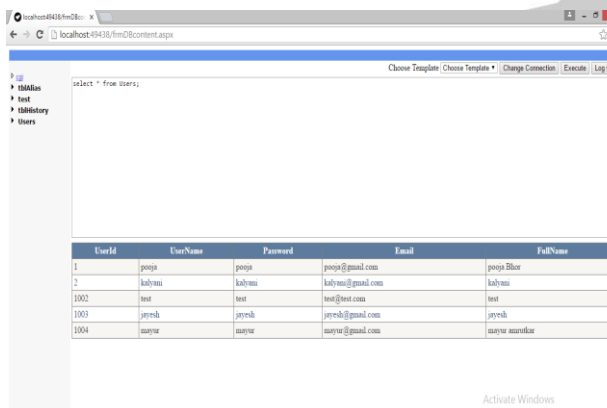
User Panel



Admin Login



User History



UserId	UserName	Password	Email	FullName
1	poopa	poopa	poopa@gmail.com	poopa Blue
2	kalyans	kalyans	kalyans@gmail.com	kalyans
1002	test	test	test@test.com	test
1003	jayesh	jayesh	jayesh@gmail.com	jayesh
1004	mayur	mayur	mayur@gmail.com	mayur sunilkar

VI. CONCLUSION

In Conclusion, We generate a web based common interface environment for multiple databases. Addressed a crucial issue in the distributed information environment: how to support effective interaction of users with heterogeneous and distributed information resources. Most studies in this problem area have focused on technical issues and performance issues but not interaction issues. The general assumption behind such studies is that combining multiple information resources will be preferred by searchers; therefore, integrated access to multiple and distributed information resources is assumed to be an appropriate mode of interaction.

ACKNOWLEDGMENT

It gives us great pleasure in presenting the preliminary project report on “Web Based Application for Accessing Multiple Databases through Common Interface.”

I would like to take this opportunity to thank my internal guide Prof. R. D. More for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.

REFERENCES

- [1] Berners-Lee T. and Cailliau R., “World Wide Web Proposal for a HyperText Project”, CERN European Laboratory for Particle Physics, Geneva CH, November (1990).
- [2] Berners-Lee T., Cailliau R., Luotonen A., Frystyk Nielsen H. and Secret A., “The World-Wide Web, Communications of the ACM”, 37(8) (1994).
- [3] Robinson D., “The WWW Common Gateway Interface Version 1.1”, Internet Draft, January (1996).
- [4] Everitt P., “The ILU Requested: Object Services in HTTP Servers”, W3C Informational Draft, March (1996).
- [5] Eichmann D., McGregor T. and Danley D., “Integrating Structured Databases Into the Web: The MORE System”, in the proceedings of the First International WWW Conference, Computer Networks and ISDN Systems 27(6) (1994).
- [6] Perrochon L., W3 “Middleware”: Notion and Concepts, Workshop on Web Access to Legacy Data, Boston, MA, December (1995).
- [7] Eichmann D., “Application Architectures for Web-Based Data Access”, Workshop on Web Access to Legacy Data, Boston, MA, December (1995).
- [8] Microsoft dbWeb 1.1 Tutorial, Microsoft Corporation (1996).
- [9] WebDBC White Paper #1, “A Quick Overview of the WebDBC 1.0 Architecture”, Nomad Development Corporation (1995).

AUTHORS PROFILE

1 st Author Name: - Jayesh Badhan

Qualification :- Diploma in Computer Engg from Mumbai University, B.E Appear of computer Engineering department from Late G.N College Of Engineering Savitribai Phule Pune University.

2 nd Author Name : - Kalyani Kanade

Qualification :- Diploma in Computer Engg from Mumbai University, B.E Appear of computer Engineering department from Late G.N College Of Engineering Savitribai Phule Pune University.

3 rd st Author Name: - Pooja Bhor

Qualification :- Diploma in Computer Engg from Mumbai University, B.E Appear of computer Engineering department from Late G.N College Of Engineering Savitribai Phule Pune University.

4 th Author Name : - Mayur Amrutkar

Qualification :- Diploma in Computer Engg from Mumbai University, B.E Appear of computer Engineering department from Late G.N College Of Engineering Savitribai Phule Pune University.

