ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (204-207), Month: January - March 2015, Available at: www.researchpublish.com

Database Switching and GUI

Er. Farida Attar[1], Mr. Shahbaz Ansari[2], Mr. Arham Charfare[3], Mr. Junaid Shaikh[4]

M. H. SabooSiddik College of Engineering, Mumbai – 008, India

Abstract: **Database Switching includes execution of all the queries on different machines connected in LAN with different backend without rewriting the queries. Database Switching between the databases reduces the work of rewriting the queries for each of the database and thus its saving time. User just has to select the Source DBMS and Table from that DBMS and a switch file is created on just a click of a button. Now this switch file can be taken to any machine (can be a remote machine) to switch data in any of the DBMS. Database GUI helps a user to do all database operations through a GUI provided with minimal knowledge of the database. Here user can do all database activities like Creating Table, Inserting Data, Joining two tables etc with minimal knowledge of query.**

Keywords: **Database Switching, GUI.**

1. INTRODUCTION

The main motivation of our work comes from experiences in using database query languages. Even people with a computer science background (ourselves included) often have difficulty using the so called "high level user friendly languages". Generally people working in the Bank or any other field having huge database may find difficulty in interacting with the application or to solve various customer queries, this might reduce the efficiency of the service provided to the customer. All the people working may not be technically sound with all the Queries. Therefore, in order to ease the employee usability as well as to increase the efficiency of the service to the customer GUI has been included in the application. The GUI feature makes the application user interface easy to use. Even a non-expert user can easily execute the complex queries which are inbuilt to learn various aspects of the Database. Many Queries or the creation of the Database can be done on the single click. These preloaded functions help to reduce the time of the User as well as makes the Management of Large Database Simple and Easy. Switching is use to generate the Database with different backend. It might take a lot of time to recreate the already existing Database again with a different backend. E.g. A Company might want to recreate the Same Sql Database into Oracle Database. This may take a lot of time and work to again create it. Switching makes the task quite easier for the user. By using the switching Application any Database can be regenerated using different backend. This saves the time and reduces the work of rewriting the complex Queries in large Database.

- Querying through GUI to execute complex as well as simple queries has proved to be an efficient system to people who do not have much knowledge about databases.
- Switching between the databases reduces the work of rewriting the queries for each of the database thus saving time.
- The Graphical User Interface generate.

2. EXISTING SYSTEM

In database switching, it is very difficult to switch records from one database to another database, so we will make to overcome this difficulty through our software.

Non-expert users may not have the patience, ability, or desire to learn and use these languages correctly. By non-expert users (as opposed to casual users), we mean non-computer science professionals such as social analysts, statisticians or accountants who have to deal with data regularly.

The problem becomes much worse in an environment with very large databases that have very large and complex database definitions (schemas). Large statistical databases such as the Census database and energy database are examples of such an environment. We believe that the following factors are the major reasons for the difficulty in using and understanding query languages.

Disadvantages of using query languages:

The user has to remember too many things. The names of the record types and attributes have to be remembered before the user can express a query. Lower level details such as the format and units of the attributes are only found by exploring the data, or looking the attribute definitions up in a dictionary (sometimes as parts of manuals). On the semantics side, the user has to determine the "meaning" of acronyms used to represent elements (record types and their attributes). All these problems are magnified when the database has a very complex schema (hundreds of record types, thousands of attributes).

RDBMS:

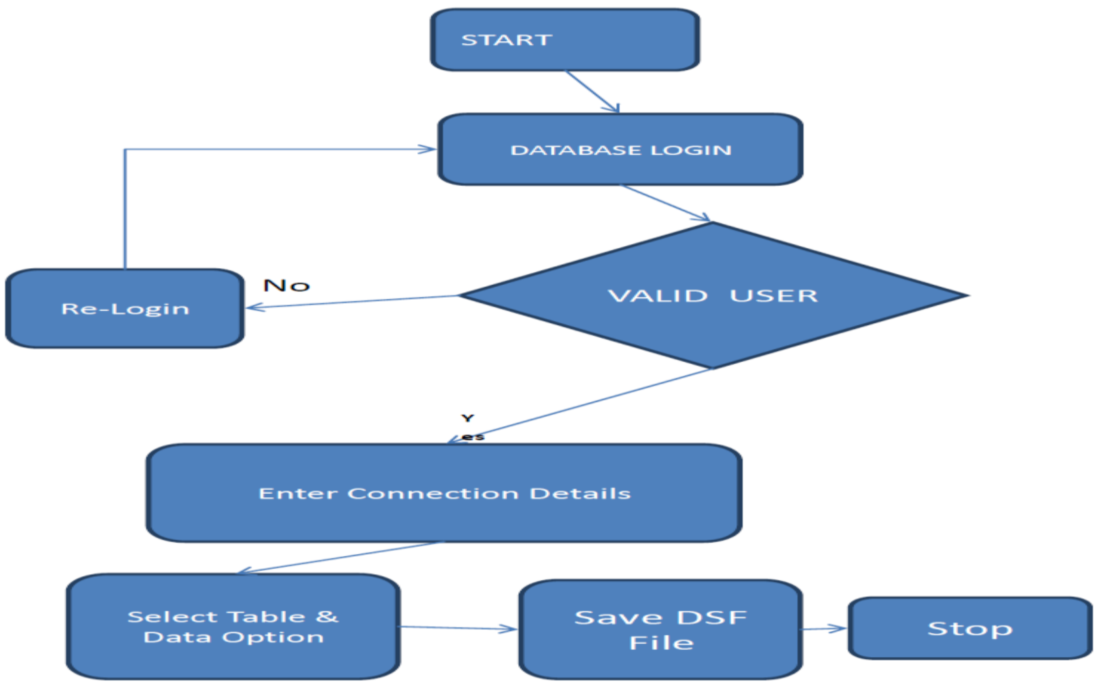
Relational Database Management System (RDBM) has been established as the technology, handling databases up to terabytes. Relational DBMSs have been extremely successful in the market. During the last two decades, Relational Database Management System (RDBM) has been established as the technology, handling databases up to terabytes. **[8]** Relational DBMSs have been extremely successful in the market; however RDBMS lack the mechanisms to deal with complex structured data

3. PROPOSED SYSTEM

Database research has concentrated on object-oriented data models, which allows to store highly structured data. With regard to the data structuring concepts offered, an object-oriented data model can be looked upon as an extension of the nested relational model, which allows storing relations as attribute values. Managing very large amounts of data[2]. In database switching, it is feasible to switch large records from one database to another database, so we will make to overcome this difficulty through our software. The main motivation of our work comes from experiences in using database query languages. Non-expert users may not have the patience, ability, or desire to learn and use these languages correctly. By non-expert users (as opposed to casual users), we mean non-computer science professionals such as social analysts, statisticians or accountants who have to deal with data regularly. The Graphical User Interface generates the queries using Oracle as the database. Similarly other databases can also be used in order to execute the queries using databases such as MySQL, Sqlserver, MS access, etc. with the GUI. This use of databases to generate the queries can be successfully implemented so that the work of writing full queries can be reduced and a non-technical user can also use it without having any knowledge of databases. User can select among the four databases such as oracle, MySQL, MS access, Sql server from which the file is to be switched and can switch the file to be executed in any of the four databases.

4. OVERALL SYSTEM DESIGN

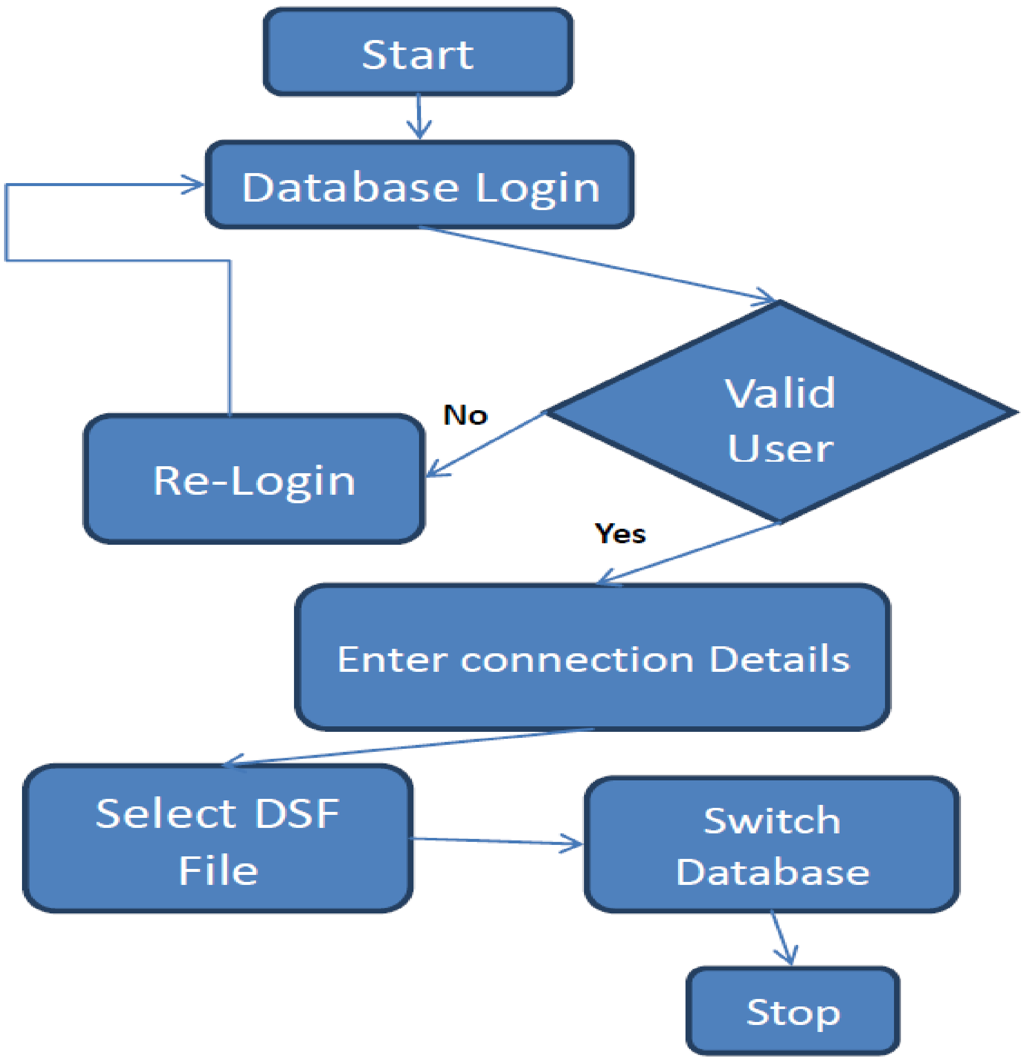
System Architecture:



ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (204-207), Month: January - March 2015, Available at: www.researchpublish.com

DATABASE GUI: In Database GUI user need to login with the database i.e Oracle. If user is a valid user with valid username and password then user will be connected with the database and if user is not a valid user then connection will not be permitted to the user. After the successful login user will be allowed to do all the operation provided in the GUI. i.e CREATE, ALTER, DELETE, MODIFY ETC.

DATABASE SWITCHING:



In Database Switching user will select the source database from which he will switch the data. After selecting the source database user will create dsf file of the data and save it to the desired location.

Then for switching data user need to select the destination database in which user desired to switch the data. Once the destination database is selected user need to select the dsf file from the saved location and switch it. Data will successfully switch to destination database

5. CONCLUSION

In contrast to a traditional setting where users express queries against the database schema, we assert that the semantics of data can often be understood by viewing the data in the context of the user interface (UI) of the software tool used to enter the data. That is, the users will understand the data in a database by seeing the labels, drop-down menus, tool tips, or other help text that are built into the user interface. Database switching allows execution of queries on different machines connected in LAN with different backend without making much effort in writing the query again. Through switching user can save much of the time in rewriting the queries.

REFERENCES

[1] www.ieee.org/publications_standards/publications/periodicals/index.

[2] J. Plodzień, A. Kraken, “Object Query Optimization through Detecting Independent Subqueries”, Information Systems, Elsevier Science, 25(8), 2000, pp. 467-490.

[3] Michel Bleja, Krzysztof Stencel, KazimierzSubeita, “Optimization of Object-Oriented Queries Addressing Large and Small Collections”, Proc. Of the IMCSIT, 2009, ISBN 978-83-60810-22-4, Vol. 4, pp. 643-680.

[4] K.Subieta, C.Beeri, F.Matthes, J.W.Schmidt. “A Stack-Based Approach to Query Languages”. Proc.2nd East-West Database Workshop, 1994, Springer Workshops in Computing, 1995, 159-180.

[5] MinyarSassi, and AmelGrissa-Touzi “Contribution to the Query Optimization in the Object-Oriented Databases” World Academy of Science, Engineering and Technology 11 2005.

[6] G. Gardarin, “Object and relational databases” Eyrolles, 1999.

[7] R.G.G. Catell, “Object-Oriented Data Management: Object-Oriented and Extended Relational Database Sytems”, Addison-Wesley Publishing, Inc., 1994.

[8] SunandaLuthra “Architecture In Object Oriented databases”Lecturer, Department of CSE/IT Amritsar College of Engg. & Tech, Amritsar.143001, Punjab, India.

[9] David Maier “Object-Oriented Database Theory An Introduction & Indexing in OODBS.