

Developing Integrated Engine for Database Administrator and Developer

Alan Seelan, Jeongkyu Lee
Department of Computer Science and Engineering
University of Bridgeport, CT
{aseelan,jelee}@bridgeport.edu

Abstract:

With the advances in the field of database there exist many tools and software that are operated separately by the developer and the administrator to perform their respective tasks . However, it is highly desired to accommodate both the administrator's and developer's roles in a database related tool, since there are many common activities between them. To address the issue, we developed the integrated database management tool for both database administrators (DBA) and developers. First, the integrated engine has various access functions to both DBA and developer to manage user instances, administer privileges and monitor databases. Second, the efficient accessing of the result set is developed in the engine without granting or revoking temporary privileges. Therefore, DBA does not have to provide the privileges to users who want to access the result set, which makes the system more secured. Thirdly, an optimized performance meter is triggered as a combination of all the important factors that determine the performance of the database, such as different hit ratios and CPU activity. Lastly, to provide developers a convenient tool for tuning SQL, we develop a PL/SQL editor where a user can parse the commands easily and perform basic activities such as a routine structure like editing, querying, optimizing (which includes hit ratio, type of scan, cardinality and selectivity). The effective outcome of the Integrated Engine is to provide a common platform to these engineers with alternative functionalities and faster accessibility towards different tasks.

I. Introduction

With highly desirable need of database and history preserving there are lots of tools that are being designed to accommodate high tactical feature with the need of the customer in mind. There are high competitive tools which provide solutions for high complex workload and easier methods to implement and access data. Functional needs of Developers and DBA's are highly integrated with the concept of providing one possible solution but they are accessible in different variations. Though there are a lot of functions that are included in much software not everything is being used in the real world so this project includes functions that are primarily used by the users.

So the major goal or objective of this project is to implement software that is standardized with the functionalities of both the developer and also the DBA's with the restriction of certain instances on the developers and also provides GUI for easy access of functionalities with also the inclusion of editor to hard code your own query and subprograms.

An Illustration of the individual user are stated, the Database Developer involves in some common activities as a routine structure like editing, querying, optimizing (which includes hit ratio, type of scan, cardinality and selectivity). On certain cases of assumption there are also other factors that may affect the execution of query based on the table size and record size. The harder workforce lies on the administrator to incline the parameter involved in effective management of the data. The module of the Administrator configures the privileges to the user and also defines a

effective approach that leads to interoperability and also just passing the result set instead of actually directly passing the user the access to the table or any database objects by doing this the user has not been granted any privilege all that is being done is the admin looks into the statement or access method the user wants to view based on the request the admin executes the query and just passes on the result set of the statement parsed and not the actual granting function more auditing tools are performed to make the sure the database performance is par the limit of maintenance and providing the statistical data analysis of the database or the instance to determine the memory hits and also the library cache information.

II. Integrated Database Management Functions:

The main objective and core concentration in this is to identify the commonalities between the developer's and the DBA's. Most of the functions that are used by the developers to code PL/SQL programs to provide an interface to the business and also some of the task to create table, table spaces, partitions and many other primary functions. So our basic aim was to develop an application instance/module to construct functions or tasks that were used by both the users.

Constructing an integrated view of the users was the basic deployment of the application. A File menu model was defined to provide some basic functionality.

- 1) SQL window/Command Window
- 2) Table, View, Sequence ,Synonym , Directory ,Job
- 3) PL/SQL functions (Package ,Procedure, Functions, Triggers)
- 4) System Functions (User ,Role, Profile)
- 5) Login Instance

The above model creation is based on GUI to ease the use of writing query and also and editor through which much complex programs or queries can be instantiated. Login instance was deployed to differentiate developer and DBA's and also providing a restriction factor on the users. The Login's are authorized with DBA's functionalities as well as developer functions based on the roles that are assigned to each user [3].

Encapsulating the features of DBA is one module of support with the features of developing and also managing will provide a high availability of instance support on a front line basis. On a DBA's perspective the approach towards the instances and objects that are created in the schema can be monitored using session logins and also provides a easy interface to find the objects associated with a schema using the object search function and also can be validated using the object valuator.

III. Proposed Way Of Administering Privileges

There are ways of representing the administrative functions of the administrator the proposed function deals with returning just the result set of the statement requested by the user instead of the granting the function to the user by this way you can still restrict the user from accessing the table and only the administrator performs the functions for the user. The reason this method of proposal was included is because that once a grant is applied to the user the user can modify any data on the table without any further reference or approval that makes the table more reluctant to change and might affect the whole user complexity of accessing the table and make evolve into more complex problem.[3]

The whole process of this kind of administrative function is categorized into three modules

- Send Request
- Check Request
Grant/Deny Request
- View Request

The Send Request is just like filling a form up which includes the job id the user that is specifying the request and the user id which are retrieved from the all_users internal table and the statement you want the admin to execute and a valid reason stating why you want to access the table. Job Id. The job Id should be specified by the user and has to be unique no two jobs should have the same ID a primary key and a not null function. Owner/Owner ID: They are by default revived from the all_users table.

Additional information of the admin when the request was submitted to perform any outdated function on the request SQL statement: The user specifies the SQL statement that the admin wants to retrieve the result set for. Reason for Request: The user specifies the reason of why the user wants to access the table only valid reason will be granted action and the rest will be ignored. The check ID button checks for the available ID.

The screenshot shows a window titled "Check_Requests". At the top, there are three buttons: "Grant", "Check Requests" (which is highlighted with a blue border), and "Deny". Below these buttons is a table with the following columns: "Job_ID", "User", "Owner_ID", and "SQL_Statement". The first row of the table has an asterisk (*) in the "Job_ID" column. Below the table is a horizontal scrollbar. At the bottom of the window, there is an "Execute Q" button followed by two empty text input fields. Below these fields is a large, empty rectangular area, likely for displaying query results or logs.

Figure 1. Check Request with Grant and Deny Function

The next update of the submitted request is processed by the administrator wherein the admin checks for the request of result set for any table access. The administrator does a check on the table for any request and depending upon the need of the user the request can be granted or denied. The check request basically does a select statement on the table where the request are inserted and then they are returned in data grid view with the all the information about the request. It also returns the reason for the request the date and time to prioritize the request and also to keep a track of request that has been received, granted or denied.

The execute statement includes the execution of the query and the user that proposes the statement for execution the user has to provide the exact function that has to be included if there is any flaw in the statement the execute statement will not be implemented so the user has to provide all the necessary objects for the correct result set needed. The above statement is executed in a grid view and if the admin finds the reason of implementation needed by the user the result set is granted for the user.

The grant/ deny function constitutes to the acceptance or rejection of the request on the user. Based on the above execution and valid reason they are granted or rejected. The grant statement doesn't do much of a work here it updates the required field into a another table which can then be viewed by the user

The view request is for the user to view the request that has been made and also view if it has been granted or not. The user can review the information until and unless the admin takes any action on the request not all the information can be viewed by the user only the information or request made by that particular user can be viewed that is exactly why the user object is used in the table to sort the field based on the updates involved in the table. So if the user wants to check whether the user has been granted / denied of the request made the user uses this form to check the status. To resend the request the user has to use the end request form again and specify the job id followed by the last job id of the request a little too complicated because the form will not allow the user to access the same job id since that does not return a null function. If the user has been granted the request the result set will be displayed in a text box with no reference of the column name if the user has not specified.

IV. Performance Meter Analytics

The performance meter is one step process of analyzing all the available information of the database and also the availability and performance of the system based on a query that was parsed against the database and also the memory usage statistics. The high activity instance can also be measured with load on the database based on this information it allows the DBA to perform more operative measures that will improve the performance of the database as well as provide a faster result retrieval.

It provides you with the information of what instance and database detailed measures with the Data Hit Ratio, Parsing Hit Ratios, Library Cache, Full table Scans, Parse Types and also metrics on the data dictionary. Volumetric Analysis can also be done basis on the information provided by the tool to manage buffer hits, Disk Reads, Execution status, Rows, sorts, loads and modules that perform these actions.

V. Basic Login and Form Functionalities

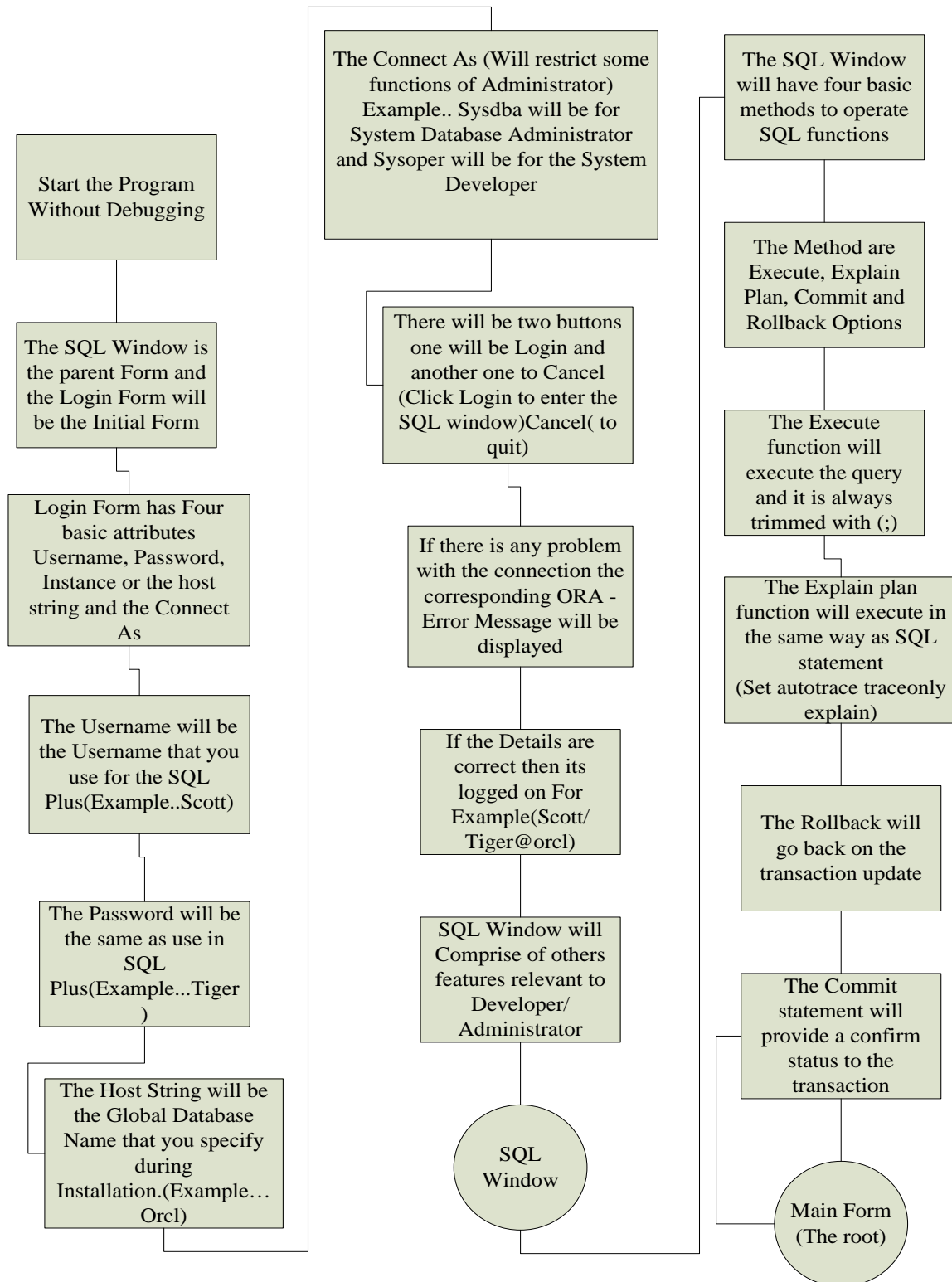


Figure 2 . Login and Form Functionality

VI. Conclusion

The project provides a platform of use for both Oracle Developers who deal with the implementation and execution of SQL statements and also for Oracle Database Administrators who deploy their work by checking for security and management of the data. The basic understanding of the project inclines it's necessity towards DBA's work force which is more complicated as well as time consuming operations are reduced.

In the future to enhance this module with various platforms of database and also providing a much interactive and display properties with more enhanced functionalities and works tasks. This is platform will be more helpful for studies and core understanding of how different measures cause inconsistency in database and system level performance.

VII. References

[1] Edward F. Gehringer and Tony M. Louca, "Using the Computer Architecture Course Database," Proc. Workshop on Computer Architecture Education, Vancouver, June 10, 2000, pp. 85–89, also in September 2000 IEEE Technical Committee on Computer Architecture Newsletter

[2] Craig.S.Mullins , "Database Administration: The Complete Guide to Practices and Procedures"

[3] Vieira, M. Madeira, H., " Towards a security benchmark for database management systems". Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on 1 July 2005.

Bibliographical Information

Alan G.Seelan

Primary Research interest in database with a major focus on the performance measure and activity of workloads on the database with resource of assigning degrees of threads to provide parallelism.

M.S in Computer Science, University of Bridgeport, USA

Phone: 618-406-5325

Email: aseelan@bridgeport.edu

Jeongkyu Lee ,Ph.D.

Assistant Professor in the Department of Computer Science and Engineering ,University of Bridgeport