IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

Database Administrator Roles and Equivalent of a SQL Server Identity in Oracle

Saeed M Alshahrani

Department of Computer Science, Sacred Heart University, Bridgeport, CT, USA

Abstract: In this paper, many tasks and responsibilities that every database administrator should and must do in order to get the perfect result for their database. There are similarity between SQL server identity, oracle identities and other database identities such as MySQL, and Microsoft SQL Server. With the 11g release the tuning advisor's technology has been fully automated so that some SQL performance problems can be fixed automatically by the system.

Keywords: Database, SQL, Oracle, Administrators, Roles.

1. INTRODUCTION

There are many tasks and responsibilities that every database administrator should and must do in order to get the perfect result for their database. Database administrator roles include the following: database design, backup and recovery, security, database performance monitoring, and other roles. I am going to explain those roles and why they are important? First, database design, designing your database will give you as a database administrator good database quality and performance because when you are designing the database you are thinking of saving the user data and how the user is going to get the data that they asking for. A good and storing database design will give the user and the application a useful database. Striping is done on two levels. The first level is the striping done by the disk arrays such as RAID, the second level is the striping of data onto separate disk arrays by DB2 UDB. On the RAID level, we make the distinction between a strip size 1 and a stripe size. On the database level, we make the distinction between an extent size and a prefect size [1]. A good rule of thumb is to make one stripe size equal to one extent size, so that the RAID striping matches the database striping. To better grasp the concepts behind striping, we will work through a disk layout example. In this example, we take an OLTP database, and place it on 8 RAID arrays, each RAID array containing 12 disks.

Second, backup and recovery, this is very important because if the application fails for any reasons it is the responsibility of the database administrator to give a copy of the recent backup that he or she has. Doing backup and recovery for the database gives the database administrators the guarantee of having the customers and users data all the time so they will not lose their users if something wrong happens to the application.

Model organism system databases (MODs) are a vital tool for scientific research. They share a common set of tasks: to collect and curate data from the scientific literature such as mutations, alleles, genetic and physical maps, and phenotypes; to integrate this information with the results of large-scale experiments such as microarray studies, SNP screens, and protein-interaction studies; to provide reagent resources such as stocks, genetic constructs, and clones; and, lastly, to provide a common nomenclature for gene symbols, anatomic terms, and other elements of the scientific vocabulary. There is similarity between SQL server identity, oracle identities and other database identities such as MySQL, and Microsoft SQL Server [2].

All those databases support the identity of auto increment and I am going to explain the equivalent of a SQL server identity in oracle and why many database administrators use it? The identity of auto-increment can be written in such different methods depends on which database that you are working with.

In oracle this identity written as SEQUENCE object which gives the database developers the ability to do their auto increment for a particular column easily where SQL server uses IDINTITY () function to do the same purpose.

II. AUTHORIZATION AND RELATIONAL DATABASES

-grained access control was first introduced as a part of the access control system in INGRES by Stonebreaker and Wong (1974), which was implemented by query modification technology. The basic idea of query modification is that before being processed, user queries are transparently modified to ensure that users can access only what they are authorized to access (Bertino et al., 2005; Wang et al., 2007).

Views are used to specify and store access permission for users. When a user submits a query, DBMS first finds all views whose attributes include the attributes of the issued query, and then add the predicates of these views to the predicates of the original query to form a new modified query, which will be carried out [4].

FGAC controls the access of users in a relational database and the access modes include select, insert, update, and delete. In relational databases, there exist mainly two approaches to granting privileges to users. One is to directly assign permissions to users, and the other is to grant privileges to the roles that users are assigned figure 1.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

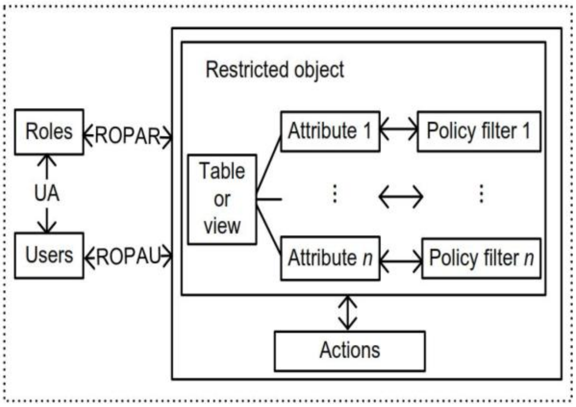

Fig1: Fine-grained access control (FGAC)

III. REFERENCE OBJECT-ORIENTED DATA MODEL

In this model a class is defined by specifying its name, its attributes, and the names of its super classes. Multiple inheritance and the existence of a default class, called TOP_CLASS, root of an inheritance hierarchy encompassing the entire database are assumed. An attribute is defined by specifying its name and its domain. Classes have both the intentional and extensional meaning and an object can be instance of only one class. An object, however, can be member of several classes through the inheritance hierarchy. Attributes can be single-values or multi-valued. In defining multi-valued attributes, the various object-oriented data models use different constructors such as set, list, tree, array. In the reference model we will abstract from specific constructors, and we assume that multi-valued attributes are defined by using a constructor denoted as set-of. The following definitions specify a notation for the Reference Model [3].

In most cases, versioned objects are shared among several users. Mechanisms should be provided so that users receive consistent and stable versions. Most version models distinguish between transient and stable versions. A transient version can be modified or deleted. However, no versions can be derived from a transient version. A transient version must first be promoted to a stable version before new versions can be derived from it. By contrast, a stable version cannot be modified. However, it can be used to generate new versions.

n the remainder of this section we describe our basic framework for database integration. In section 2 we discuss attribute equivalence and "embedded" attributes, and we introduce the "locality attribute", a central concept in our integration methodology. In section 3 we define the "two-dimensional union" (our basic integration operator), we show how the "top-down" distributed database concepts of fragmentation and allocation can be introduced into the "bottom-up" integration context, and we touch briefly on questions of constraint integration. In section 4 we describe how to translate global updates in our approach. Section 5 contains conclusions and suggestions for future work in this area.

IV. EMBEDDED CALCULATED ATTRIBUTE

For this category, the value of the embedded attribute is calculated as a function of other attributes. A special case of this situation was mentioned by Kent in, where the term "embedded attribute" was used to refer to a situation in which a character string attribute contained several properties of independent interest literally embedded within it as substrings. As another example, suppose we have a relation which describes individual automobiles in a database at an automobile production site, and suppose that the interior color for automobiles produced at this site is always the same as the exterior color. In this situation, the designer of the database might feel that it is redundant to include both exterior color and interior color as attributes in the relation, leading to the relational scheme. There are similarities between the concept of embedded calculated attributes and the concept of functional dependencies, but there are also important differences. If there exists a functional dependency X + A, it only means that the value of the attribute A is a function of the values of the attributes in X at each point in time. The function may vary at different points in time. If the value of A is missing for a particular tuple, it can only be computed from the values of the attributes in X if there happens to be another tuple in the relation with the same values for the attributes in X. For an embedded calculated attribute B, there must be a time-invariant function allowing the calculation of the value of B at any time from the values of the attributes it is dependent upon [7].

V. TJ-II REMOTE PARTICIPATION SYSTEM CHARACTERISTICS

The TJ-II RPS is based on web servers as mentioned above. This allows one to build a very scalable and distributed environment, even avoiding direct communication between users and data acquisition systems (DAS). Such indirect communication adds security to the system while also saving DAS resources. Thus the DAS controlling computer can be entirely devoted to data acquisition and analysis tasks, rather than providing capabilities such as access control, database support, or user interface. The TJ-II RPS authentication and authorization system has to provide resources to accomplish the signal and module. ownership requirements for programming purposes [6]. All authenticated users can access, in read-only mode, all channel setups but only authorized users (signal owners) can change parameters. Finally, the TJ-II RPS must guarantee the uniqueness of signal location in order to meet this general requirement of the TJ-II data acquisition system figure 2.

From the TJ-II operation point of view, this is the main process. Users can visualize all signal acquisition parameters while signal owners can modify a channel setup. The client tier for this process consists of web pages. Several web pages, depending on the hardware characteristics (sampling rates, memory, pre-trigger and post trigger protocol and so on) of the digitizer to which the signal is associated, provide user interface [8].

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

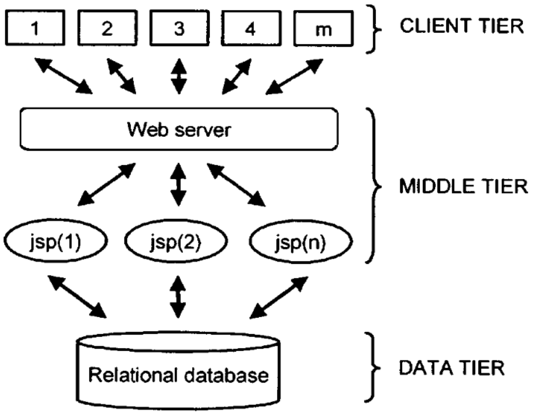

Fig 2: Multi-tier software architecture for programming

In the first stage, the acquisition cards available for the TJ-II RPS will be two families of in-house VXI cards, PXI 6070E cards from National Instruments, PCIDAS 4020/12 cards and PCI-DAS 1602/16, both from Measurement Computing. Over 1000 channels will be managed in this way from the TJ-II remote participation system. In addition to the hardware characteristics pages, others have been developed in order to provide users with interfaces for defining signal parameters such as amplifier gains, calibration factors, or other experimental values. There is no restriction on the number of parameters per signal, as this information is stored in the TJ-II database, which also provides a buffer for this data. Web pages devoted to channel setup exchange typically 50 Kbytes between client and middle tiers [5].

TJ-II operation can be easily followed through “TJ-II virtual system.” A virtual system can be visualized as a signal container. A user defines a virtual system through the use of a name. The user “places” signals in the virtual system and a web page provides a graphical user interface that pro- vides a constant update of the status of the virtual system when required.

VI. WORKLOAD TUNING STRATEGY

The SQL Tuning Advisor follows a statement-at-a-time model. When analyzing a SQL statement from a workload, the advisor focuses on each individual statement in isolation and attempts to find the root cause of any performance problems it might have. The four types of analysis from part A are invoked independently for each SQL statement, and any findings made are attached to the relevant SQL statement. The statement-at-a-time architecture is particularly suitable to the SQL Tuning Advisor since its goal is to fix spot problems on a production system, giving a DBA tools like the SQL profile to fix one statement without impacting others. Statement-at-a-time has the additional advantage that it scales very easily to large workload sets. The advisor avoids the potential disadvantages of this approach by summarizing findings across the workload and caching the results of key computations. When the advisor repeats findings across statements, it presents them in a workload-level summary to help DBAs fashion a high-level understanding of the tuning results. Repeated work is avoided by caching results of expensive work performed for one SQL statement and re-using it for similar statements later. A good example of this is the data sampling performed by SQL profiling, where table statistics are computed once and reused across statements.

As a component that requires detailed interactions with the end user, a powerful reporting infrastructure and graphical user interface (GUI) have always been essential pieces of the SQL Tuning Advisor. The advisor provides a detailed textual report to command-line users and an interactive GUI in Oracle Enterprise Manager (OEM) where it gives a clear explanation of its advice for each particular SQL statement [9].

VII. CONCLUSION

Simplify SQL performance tuning; Oracle introduced the SQL Tuning Advisor in 10g. It made many performance issues much easier to fix, but was lacking an automation framework: users still had to run the advisor manually when performance problems happened and accept the advisor’s recommendations. With the 11g release the tuning advisor’s technology has been fully automated so that some SQL performance problems can be fixed automatically by the system. It provides safeguards to guarantee that its tuning activities will not harm any application workload. One natural question which arises is whether the "two-dimensional union" is flexible enough to handle the complex situations which arise in integrating actual databases. We believe that its ability to integrate relations which represent essentially the same type of objects, even though they may have different attributes, does provide a great deal of flexibility, but this has not yet been demonstrated in the context of a real-world application. This is a good question for further work.

REFERNCES

[1] G.D. Held, M.R. Stonebraker, and E. Wong, "INGRES – A Relational Database System," Proc. AFIPS Nut. Computer Conf., Vol. 44, AFPS Press, Arlington, VA, 197.5, pp. 409-416..
[2] B. Dageville, D. Das, K. Dias, K. Yagoub, M. Zait, M. Ziauddin, “Automatic SQL Tuning in Oracle 10g,” in VLDB pages 1098-1109, 2004..
[3] Al-Kahtani, M.A., Sandhu, R., 2004. Rule-Based RBAC with Negative Authorization. Proc. 20th Annual Computer Security Applications Conf., p.405-415. [Doi: 10.1109/ CSAC.2004.32].
[4] Bertino, E., Samarati, P., Jajodia, S., 1997. An extended authorization model for relational database. IEEE Trans. Knowl. Data Eng., 9(1):85-101. [doi:10.1109/69.567051]
[5] Lee, C. C. (1990). Fuzzy logic in control systems: fuzzy logic controller. II. Systems, Man and Cybernetics, IEEE Transactions on, 20(2), 419-435.
[6] Ragot, J., & Lamotte, M. (1993). Fuzzy logic control. International journal of systems science, 24(10), 1825-1848.
[7] Rizvi, S., Mendelzon, A., Sudarshan, S., Roy, P., 2004. Extending Query Rewriting Techniques for Fine-Grained Access Control. Proc. ACM SIGMOD Int. Conf. on Management of Data, p.551-562. [doi:10.1145/1007568.1007631].
[8] Olson, L.E., Gunter, C.A., Cook, W.R., Winslett, M., 2009. Implementing reflective access control in SQL. LNCS, 5645:17-32. [Doi: 10.1007/978-3-642-03007-9_2]
[9] Kuznik, F., Virgone, J., & Roux, J. J. (2008). Energetic efficiency of room wall containing PCM wallboard: A full-scale experimental investigation. Energy and buildings, 40(2), 148-156.