https://doi.org/10.31871/WJIR.8.6.12

World Journal of Innovative Research (WJIR)
ISSN: 2454-8236, Volume-8, Issue-6, June 2020 Pages 101-107

Application of Database Auditing for Students' Academic Records

Abdulrahman Yusuf

Abstract — **A database holds essential assets of an organization. Students' information, customers' information, employees' information and employers' information of an organization are being kept in the databases. There has been an increase in reported occurrences of data abuses especially by insiders. This research work aimed to investigate for appropriate database auditing techniques applicable on students' academic records and present logical steps and procedures for implementation using SQL/PLSQL programming in Oracle 11g. Yobe State University (YSU) was used as a case study. Trigger-based auditing, fine-grained auditing, auditing SYS users, and audit trail management are found to be appropriate techniques implemented and evaluated in the proposed developed YSU system. In addition, database security in term of careful creation of database users, privileges allocation to the users, and auditing the activities of the users are also found to be things of significant concern for the reliability and integrity of data. Empirical evaluation showed that application of database auditing on students' academic records could check proud and internal threats imposed by insiders.**

Index Terms— **Database Auditing, Security, Students' Academic Records, Insiders, and Oracle 11g.**

I. Introduction

The proliferation and the use of modern technology made governmental and nongovernmental organizations accommodate the use of computers for the collection, keeping, and sharing of data and information electronically. Therefore, there has been an increase in reported occurrences of data abuses, especially by insiders. As such, data need to be secured for reliability and integrity. Database auditing is an integral part of database security and can be used to proctor the activities of database users.

Data is the bedrock of any organization and students' academic records appear to be the most valuable asset to the educational institutions. According to [1] " … *databases used to store sensitive information are now the target of numerous regulations requiring accountability for how data is handled. To meet this challenge, organizations should implement strong database auditing practices.*" The objective of this research is to investigate appropriate database auditing techniques and develop an application that could serve as a guide to developers or security administrators responsible for auditing users' activities particularly in a students' academic records database.

Trigger-based, fine-grained, SYS users auditing, and automatic audit trail management are considered to be the most appropriate techniques for enforcing security in the students' academic records database. The students' academic records system at Yobe State University (YSU) developed, has been used to demonstrate the design, implementation and reliability of these appropriate auditing techniques.

A. *Statement of Problems*

Databases for students' academic records appear to be the most valuable asset to the educational institutions. Therefore, they should be secured against internal threats for integrity and reliability using appropriate database auditing techniques to know who did what, when and how? In this connection therefore, there has been an increased in a report that students' academic records have been altered by insiders (such as level coordinators, record officers, etc) in most Nigerian tertiary institutions. Application of proper database auditing techniques on students' academic records could deal with such threats. The study used YSU for practical application using Oracle 11g.

B. *Significant of the Study*

The significances of this research work are outlined below:

1. It would generally disclose the importance of a database auditing at a glance for security enforcement on a database that holds the essential information of an organization.
2. It would present how to properly assign role-based access control for better monitoring of users' activities.
3. It could serve as a guide to both application developers and learners especially students of computer science on how to properly apply database auditing not only on students' academic records. Hence, this work will show how to:
   - i. discover suspicious activities;
   - ii. notify for users' illegal actions to the appropriate person concerned (for example, a database administrator);
   - iii. proctor and gather information about specific database activities.

II. REVIEW OF RELATED WORKS

A. *Database Auditing Issues*

Insider threats, identity theft, and corporate governance and compliance are the major auditing security issues [2]. These issues are what trigger for applying appropriate auditing techniques in a database. *Insider* is an authorized personnel in the organization having a certain level of system database

Abdulrahman Yusuf, Department of Computer Science, Yobe State University, Damaturu, Yobe State, Nigeria

access. For example, the CSI/FBI survey has shown that more than 70% of data loss and attacks were by insiders [3]. *Identity theft* is directly related to information stealing in an enterprise or organization. Governments and businesses have been experienced potential loss of data as a result of identity theft, therefore, they use database auditing as an alternative measure [2]. For corporate governance and compliance, many laws have been promulgated by the government to protect the information and data of investors, patients, customers and citizens, therefore auditing is used for compliances [2], [3], [4].

The prominent databases that provide native database auditing tools are Oracle, Microsoft SQL Servers, Sybase, and IBM DB2 [5]. Each of these databases has its level of auditing capabilities which can be configured and managed by personnel of an organization such as DBA or security administrator. Oracle has been rated as the top native auditing capabilities provider because it has consolidated, versatile, flexible and most secured auditing capabilities than others [5], [6] & [7]. Therefore, the Oracle database is decided to be used for this research work. The implementation and management of any auditing capability are directly involved in the use of access control.

B. *The Role of database Access Control*

In information management systems, permission to access or not is of primary importance and concern. Access permission and control of database users do provide protection to data resources from being accessed legally or illegally by legitimate or illegitimate users [8], [9] & [10]. Role-based access control (RBAC) is a major and concrete mechanism for the implementation of security measures needed by any database security administrator [10], [8]. Therefore, the concepts of user, role, privilege and object are essential to understand. Users and permissions are segregated using the notion of a role. The *Role* is created based on the collection of functionalities related to the job or qualifications of users [10], [9] *Permission or privilege* is the right assumed to perform the obligatory job functions based on a role [10] [11]. However, in Oracle, privilege can be given to the user directly rather than role but it is not ideal [11]. Access to an object by a user should be based on role. A group of users, with the same assignment or work to execute in an organization are placed on a role. Resources are accessed based on privileges associated with the roles. Therefore, resources or objects have a direct relationship to the roles rather than the users as depicted in **Fig. 1**. Oracle categorized the privileges into two: *System privileges* and *Object privileges*. In *System privilege*, permissions are given to execute DDL statements or permissions are given to grant or revoke the privileges to/from the users or roles [11], [12]. *Object privileges* permit the execution of a given action on a particular object such as a table, view, sequence, procedure or function [11], [10]. According to [13] Oracle database is developed with the notion of objects and objects are *"all data"*. It is based on the given privileges on the objects that a user can access. The most popularly known objects are tables, some others that might not be well known as objects include roles, users or packages.

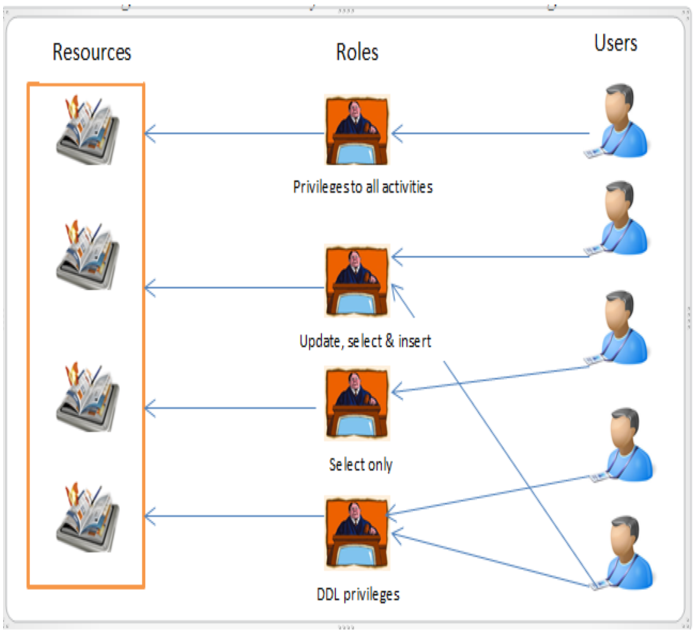

Fig. 1: Role-based Access Control

C. *The Need for Database Auditing*

[14] states the main facets of security as authentication/identification, authorization, and auditing. [15] added encryption. These facets could only be attained with the help of access control explained above. The facets are put in place hierarchically. The authentication is the solid foundation to pass by the user first, then authorization to identify the user's privilege. As such, users' activities can be monitored after authentication and authorization based on the defined policies. These layers of hierarchy are in co-existing; a serious problem may be resulted in the whole security measure if one layer is affected [15].

Database auditing techniques contribute significantly to the planning and implementation of database security for ensuring information security system [15], [16]. [11] states that for both forensic analysis and regulatory compliance with data/information protection laws, database auditing has become a tool of considerable importance. Generally, database auditing is typically carried out to:

- discover suspicious activities;
- notify for users' illegal actions to the person concerned (for example, a database administrator);
- proctor and gather information/data about specific database activities and
- handle and comply properly with auditing requirements.

By considering the roles play by database auditing for ensuring security in an information system and especially for data/information integrity, this research will use auditing techniques to find better ways of providing security and preserving the integrity of students' academic records particularly at YSU Nigeria. .

D. *Overview of Some Auditing Techniques*

Some related auditing techniques in oracle 11g are categorized and analyzed in Table 1.

Table 1. Related auditing techniques of Oracle 11g

Related Techniques	Description
1. Application auditing	It is implemented in a third party application not in an actual database. Although application auditing is flexible and portable through the DBMS, it requires much more efforts to maintain and can be easily compromised or bypassed [14].
2. Standard auditing	Statement auditing, privilege auditing, auditing connection and schema object auditing fall under standard auditing categories. [17] name them auditing types. Generally, objects, users, system privileges, succeeded/non-succeeded actions and procedures execution are what to specify and audit under the appropriate standard auditing category [14]. [14] further states that standard auditing generates so many records, therefore, resulted in performance impact and storage issues; and audit condition cannot be placed based on specific conditions or columns.
3. Trigger-based auditing	Trigger can maximize data/information integrity of an enterprises and both DML and DDL transaction can be audited [18], [13]. [19] [14] explain that trigger has the provision of applying audit at the raw level; and it can be used to capture certain details before and after SQL statement execution [11], [13]. Although it can be used to capture data from the associated table, the triggering statement cannot be captured and every object requires its own copy. The user with SYS privilege is a threat to triggers capabilities [11], [13], [14].
4. Fine-grained auditing	It allows a security policy to be applied at the granular level [11], [18]. Therefore, the lower granularity application of auditing makes Fine-grained auditing to be more specific and precise. For example, audit access to an object after working hours, audit access to a sensitive column other than particular user.
5. Auditing SYS users	This permits SYSDBA and SYSOPER user important activities (such as shutdown, logoffs, insert and update) in the database to be audited usually into OS files [11], [14]. It can be achieved using script: ALTER SYSTEM SET AUDIT_SYS_OPERATIONS=TRUE SCOPE=SPFILE; Therefore, SYS auditing can be the solution to the SYS users' compromises.

Going by the above analysis and the aim of this research work which is about auditing students' academic records, it has been observed that the auditing techniques to apply need to be for specific columns and conditions. Some information from the associate tables needs to be in place for proper actions. Also, to avoid compromises from SYS users, their actions also need to be audited.

A. Managing Audit Trail

Information of operations audited, information of operation users and timestamp of operations are usually what made up of the audit records [11]. Where the audit records are kept is subject to be filled, therefore, it requires proper maintenance for better utilization. Operating system audit trail and database audit trail are two rudimentary destinations of audit records [11]. When a destination is filled-up, an error message will be resulted and a record is no longer being accepted. Therefore, the destination needs to be controlled.

In controlling the destination, the audit options should be carefully selected as well as AUDIT ANY system privilege allocation. In addition, a periodic clean-up need to be performed. The audit trail can be managed manually or automatically. Oracle recommended the use of automatic management using the procedure DBMS_AUDIT_MGMT available in Oracle 11g release 2 [20].

III. MATERIAL AND METHODS

This research has been approached using both qualitative and quantitative methods for triangulation. The qualitative information reviewed has given the basis for empirical implementation and evaluations of the proposed system developed for auditing users' activities in students' academic records which have been produced based on design, software, SQL and PL/SQL languages and policy creation.

A. Design

The design and implementation of the proposed system were approached by creating the relational database schema of YSU students' academic records after the necessary data have been obtained and analyzed from the University staff and personal knowledge of the researcher as a member of staff. This made appropriate auditing techniques investigation possible.

B. Software

The development of the proposed system has been carried out using Oracle 11g express edition which is free for development, deployment and distribution. The creation of the relational database schema, insertion of test data and creation of audit policies and conditions were using SQL and PL/SQL languages which are the fundamental instruments for any Oracle application development.

C. SQL and PLSQL Programs Tables

The SQL and PL/SQL languages facilitated the developer interaction with the Oracle database server. The SQL language allowed for the creation of schema tables and modification plus querying the database. PL/SQL language enabled for the creation and call of triggers, storage of procedures and functions as well as invoking of PL/SQL packages. SQL*Plus is a command line tool allowed for SQL and PL/SQL commands editing including query results formatting and options setting.

D. Security Policy Creations

In general, database security implementation should be database-centric rather than application-centric [14], [21]. Trigger-based, fine-grained and `SYS` users auditing are database-centric and seem to be more appropriate techniques for enforcing security measures in a students' academic records database. Using trigger-based and fine-grained auditing, security policies and conditions on objects could be defined at the granular level. **Fig. 2** depicts the general working principle of the proposed audit system.

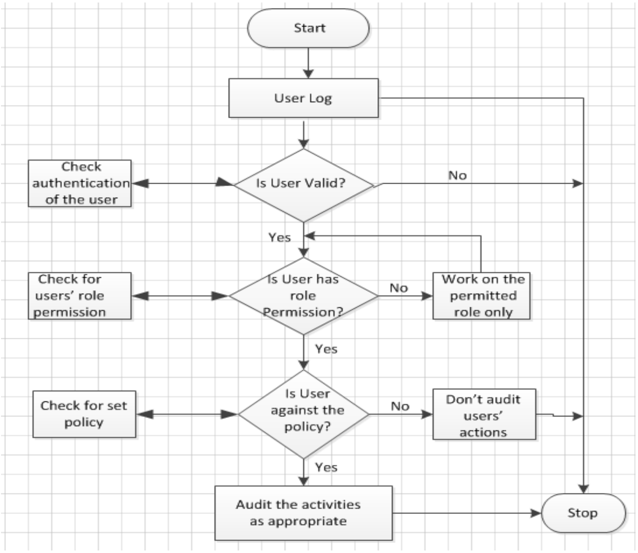
Start
User Log
Check authentication of the user
Is User Valid?
No
Yes
Check for users' role permission
Is User has role Permission?
No
Work on the permitted role only
Yes
Check for set policy
Is User against the policy?
No
Don't audit users' actions
Yes
Audit the activities as appropriate
Stop

Fig. 2. Working principles of the proposed audit system

IV. RESULTS AND DISCUSSIONS

Investigating the appropriate auditing methods and techniques for enforcing security measures in the proposed system developed of students' academic records is the main goal of this research. Auditing issues, methods and techniques will be evaluated both theoretically and empirically based on the reviewed literature and the schema designed and implemented for YSU students' academic records respectively.

A. Theoretical Approach

The continued monitoring of activities in a database in terms of 'who did what, to which data when and how?' is known as database auditing [1], [14], [11], [16] [21]. The main challenges that trigger database auditing are insider threats, identity theft and corporate governance and compliance; these are called issues for database auditing [2]. Insider threats and corporate governance and compliance seem to be of greater concern than identity theft for implementation of auditing techniques in students' academic records when compared with the need in enterprises such as banks. Laws have been formulated and refined in developed countries for data/information protection [2], [3], [4]. Auditing is used for analysis and reporting in compliance with the laws. However, in developing nations, data protection laws may not be in place. For example, in Nigeria, it is only recently that legislators started work on data protection bills [22]. It may be possible that academic institutions have local means of data protection. Therefore, this research could reveal a viable means of data protection compliance for educational institutions.

Some categories of auditing techniques are built-in and usually audit the general activities in the database when they are enabled or configured [11], [14]. For example, statement auditing, object auditing, and privilege auditing which are called standard auditing. Auditing of users' activities in students' academic records database systems needs not be extensive or wide ranging, rather it should be of specific activities that pose security threats to the records and the integrity of the institution. Based on the reviewed literature, auditing specific activities in students' academic records can be performed with the auditing techniques such as trigger auditing or fined-grained auditing or a combination of the two in some situations. Although trigger auditing can be compromised by rollback if autonomous transactions are not used, users with `ALTER TABLE` privilege can also be a threat to the records, it has the advantage of capturing information from associated tables. The use of fine-grained auditing for auditing specific activities allows certain policies to be defined at the lower levels of granularity and it allows real-time notification of undesirable activities in the database. However, users with `SYS` privilege could be a very serious threat to the audited records or security in general. To overcome the compromises posed by `SYS` users, auditing `SYS` user technique can be configured.

B. Experimental Approach

In order to support the identified appropriate auditing methods and techniques for students' academic records from the literature, the researcher has designed, implemented, and tested the auditing techniques on the YSU students' academic records relational database. **Fig. 3** shows the YSU students' academic records Entity Relationship Diagram (ERD). Triggers are created to ensure that any entry into the columns STU_ID and COURSE_CODE of the link entity `Enroll` must be from `Student` and `Course` entities respectively.

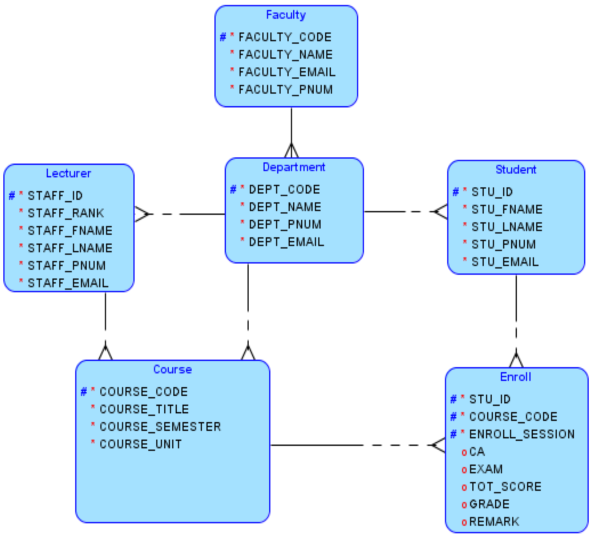

Fig. 3: ERD of YSU Students' Academic Records

1. **Database Users Creation** - It was understood in the literatures that the careful creation of database users and roles allocation provide an easy means of database and security management. Therefore, ad_officer is created to insert, select and update records on th Enroll table in some columns (STU_ID, COURSE_CODE and ENROLL_SESSION) within a specific time of enrolment. The users (mth_lecturer1 and cs_lecturer1) are examples of lecturers created to update some columns (CA, EXAM, TOT_SCORE, GRADE and REMARK) on the Enroll table within a specific semester. The user ex_officer is from the exam office for report generation, for example, transcripts issues

2. **Trigger-based auditing Techniques**

Trigger_test one: trigger audit was created to capture the changes to the students' grades, who performed the changes, and when. The capturing should be done only after the University senate approves the result, for example, on 12th July, 2013. Some statements were issued and audited as below:

```
STU_ID           COURSE    SEM_SESSION  OLD_GARDE NEW_GRADE USER_NAME          ALT_DATE
--------       --------- --------     ---------    -------   -------          ---------
U/MTH/010/001   MTH 111   2010/11      F             C       MTH_LECTURER1    06-AUG-13
U/CS/010/003   CS 111    2010/11      C              B       CS_LECTURER1     06-AUG-13
```

Each of the above records was lost after rollback, but when autonomous transactions were used the records remained even after rollback

Trigger_test two: this was created to audit any grade changes on the Enroll table after a specific date (for instance, 12th July, 2013). The audit details should be put into the audit table, and should include a course tutor name, tutor's phone number, department name, course code, the student ID, database user that makes the changes and the date. The records should never be erased from audit table even after rollback. After two transactions were performed the audit table was queried and shown as

```
LECT_NAME    LECT_PNUM  DEPT_NAME    STU_ID   COURSE_CODE EN_SES   OLD NEW USER_NAME    ALT_TIME
--------     --------- -------      -----    ---------   ------- --- ---  ----------    -------
MAMI GARBATU  9000005   MATHEMATICS U/MTH/010/001 MTH 111 2010/11  F  A  MTH_LECTURER1 06-AUG-13
LAMI MA'AZU   9898837 COMPUTER SCI. U/CS/010/003  CS 111  2010/11  C  B  CS_LECTURER1  06-AUG-13
```

Therefore, trigger-based auditing is observed to:

- allow having certain record and column details, capturing before and after executing SQL statement on the Objects;
- provide complete flexibility and simplicity around the scope and criteria and
- allow for information to be captured from other associated tables.

Trigger-based auditing could be an excellent choice for an institution, if capturing triggering statements and sending an email alert for wrong activities are not its priority. In addition users with ALTER TABLE or SYS privileges are serious security threats to the system. However, fine-grained as well as SYS users auditing technique could provide the solution.

3. **Fine-grained auditing technique**

fga_policy one: a policy was created to capture any action of admission office beyond these columns (STU_ID, COURSE_CODE and ENROLL_SEESION) on Enroll table.

```
SELECT    db_user,    timestamp,    sql_text    FROM
dba_fga_audit_trail      WHERE      policy_name     =
'FGA_ADMI_ENROL';
```

```
DB_USER     TIMESTAMP    SQL_TEXT
---------   ----------  -----------------------
AD_OFFICER   11-AUG-13    SELECT * FROM system.Enroll
AD_OFFICER   11-AUG-13    INSERT INTO system.enroll
                          VALUES('U/MTH/010/021',
                          'MTH 111', '2010/11', NULL,
                          NULL, NULL, NULL)
```

Out of six statements issued by the admission office only two were audited as shown above because they referenced other columns than those allowed.

fga_policy two: a policy was created to monitor and audit any update or insert in the columns STU_ID, COURSE_CODE, or ENROLL_SESSION by a user who is not from admission office.

```
SELECT db_user, timestamp, sql_text FROM
      dba_fga_audit_trail WHERE
      policy_name =
      'FGA_NOT_ADMI_ENROL_OFIS';
```

```
DB_USER      TIMESTAMP   SQL_TEXT
----------- ---------   -----------------------------------------
MTH_LECTURER1 18-AUG-13  UPDATE system.enroll SET stu_id =
                         'U/MTH/010/003' WHERE STU_ID =
                         'U/MTH/010/001' AND ENROLL_SESSION =
                         '2010/11'
NEW_DBA       18-AUG-13  INSERT INTO system.enroll(STU_ID,
                         COURSE_CODE, ENROLL_SESSION)
                         VALUES('U/MTH/010/001','MTH 121', '2010/11'
```

The records above were audited from the three statements issued by a lecturer, DBA and the admission officer. This is because the insert and update of the columns are not allowed by any user except the admission office.

fga_policy three: a policy was created to audit any insertion or updating by admission office after the period of admission and enrolment (for instance, after 12 – FEB-2013).

```
SELECT db_user, timestamp, sql_text FROM
```

```
      dba_fga_audit_trail WHERE policy_name =
      'FGA_OUT_OF_SCHEDULE_ADMISSION';

DB_USER     TIMESTAMP    SQL_TEXT
--------    ---------    -------------------------------------------
AD_OFFICER 11-AUG-13     INSERT INTO system.enroll(STU_ID, COURSE_CODE,
                         ENROLL_SESSION)VALUES('U/MTH/010/001','MTH
                         211', '2010/11')
AD_OFFICER  11-AUG-13    UPDATE system.Enroll SET ENROLL_SESSION =
                         '2010/11' WHERE stu_id = 'U/MTH/010/001' AND
                         COURSE_CODE = 'MTH 111'
```

Three statements of insert, update and select were issued but only that of insert and update were audited as applied in the policy and were shown from the audit records above.

fga_policy four: the assessment columns (CA, EXAM, TOT_SCORE, GRADE and REMARK) of enroll table were to be updated or inserted only by lecturers. A policy was created to send an email alert for any activity by a non-lecturer user.

```
Attention!! OPS$1223793.enroll table
violation using statement by: ad_officer1,
UPDATE OPS$1223793.enroll SET grade = 'C'
WHERE stu_id = 'U/MTH/010/001' AND
ENROLL_SESSION = '2010/11' AND COURSE_CODE
= 'MTH 111'. The time is: TUE 10 SEPT, 2013
12:05:34
```

An email alert was received as above, after a lecturer and non-lecturer users executed the same statement. This was tested using university database as it could not be possible using local-host.

fga_policy five: a policy was created to audit any insertion or updating in students' assessment columns (CA, EXAM, TOT_SCORE, GRADE and REMARK) on enroll table after the results have been approved by the University senate (for example, after 12 –JULY- 2013).

```
SELECT db_user, timestamp, sql_text FROM dba_fga_audit_trail
       WHERE policy_name = 'fga_out_of_schedule_assessment';

DB_USER        TIMESTAMP     SQL_TEXT
------------- -----------  ------------------------------------------
MTH_LECTURER1 11-AUG-13     UPDATE system.enroll SET grade = 'C' WHERE
                            stu_id = 'U/MTH/010/001' AND ENROLL_SESSION =
                            '2010/11' AND COURSE_CODE = 'MTH 111'
MTH_LECTURER1 11-AUG-13     UPDATE system.enroll SET EXAM = 35 WHERE stu_id =
                            'U/MTH/010/001' AND ENROLL_SESSION = '2010/11'AND
                            COURSE_CODE = 'MTH 111'
MTH_LECTURER1  11-AUG-13    UPDATE system.enroll SET EXAM = 50.05 WHERE
                            stu_id = 'U/MTH/010/001' AND ENROLL_SESSION =
                            '2010/11' AND COURSE_CODE = 'MTH 111'
MTH_LECTURER1  11-AUG-13    UPDATE system.enroll SET remark = 'PASS'
                            WHERE stu_id = 'U/MTH/010/001' AND ENROLL_SESSION
                            = '2010/11' AND COURSE_CODE = 'MTH 111'
```

Four statements issued by the lecturer were audited after the approval date as shown above.

fga_policy six: a policy was also created to monitor and audit any transaction on the Enroll table outside working days and hours, i.e Saturday, Sunday and between 8am and 5pm.

```
SELECT db_user, TO_CHAR(timestamp, 'DY DD-MON-YY
HH24:MI' ), sql_text FROM dba_fga_audit_trail WHERE
policy_name = 'FGA_OUT_OF_SCHEDULE_ACCESS';

DB_USER        TIMESTAMP                 SQL_TEXT
---------      ---------                 ------------------------------
CS_LECTURER1   THUS 22-AUG-13 00:08      select * from system.enroll

MTH_LECTURER1  SUN  25-AUG-13 17:09      UPDATE system.enroll SET grade = 'A'
                                         WHERE stu_id = 'U/MTH/010/001' AND
                                         ENROLL_SESSION = '2010/11' AND
                                         COURSE_CODE = 'MTH 111'
```

Only the above two statements were audited out of a number of statements issued within and outside the working days and hours.

Fine-grained auditing was found handled the failure of trigger-based auditing to: audit triggering statements; apply a policy affecting more than one column within a single trigger and define and transmit an email about policy violation.

4. SYS User auditing technique

SYS users auditing was enabled to track the SYS users' activities in the database, for their pose security threats. It was configured as

```
ALTER SYSTEM SET AUDIT_SYS_OPERATIONS =
TRUE SCOPE = SPFILE;
```

After enabling sys users auditing, the sys user had connected and executed a statement as

```
DELETE FROM DBA_FGA_AUDIT_TRAIL WHERE
DB_USER='NEWDBA'
```

The sys user activity was audited into the event log as shown below.

```
Audit trail: LENGTH : '219' ACTION :[54]
'DELETE FROM DBA_FGA_AUDIT_TRAIL WHERE
DB_USER='NEWDBA'' DATABASE USER:[1] '/'
PRIVILEGE :[6] 'SYSDBA' CLIENT USER:[14]
'yusuf-PC\yusuf' CLIENT TERMINAL:[8]
'YUSUF-PC' STATUS:[1] '0' DBID:[10]
'1345592189'.
```

5. Audit Trail Management Technique

After careful privileges allocation and audit options selection, it was attempted to use the recommended audit trail management procedure DBMS_AUDIT_MGMT available in Oracle 11g release 2 to show the better management of audit trails. The fine-grained audit trail records clean-up was initialized to be carried out after every 24 hours; the clean-up was to affect the records accumulated on or before 21st August 2013 11pm; and the actual clean-up was executed.

V. Conclusion

In this paper, a careful means of developing an application for auditing students' academic records has been presented using Oracle 11g database. The research investigated and found the most appropriate auditing methods and techniques such as trigger-based, fine-grained and SYS user auditing for students' academic records as well as their logical step by step design and implementation pattern that could serve as a general guide to application developers and database security administrators of institutions, hence saving both time and efforts involved in designing and developing from scratch. The details capability and incapability of every technique were given for reasonable selection.

Some limitations have been observed for the developed system in this work. Firstly, the idea of views was not used for users to view only their appropriate working columns, for instance, the lecturers from various departments to see and manipulate only their relevant courses. Secondly, the implementation could have used the latest Oracle release. Therefore, further work is required to make the system more reliable, secure and up-to-date.

Acknowledgment

My sincere gratitude to Allah (SWT) for the given opportunity. I am acknowledging the efforts of Dr. Marry Gervey and Prof. Carl of University of Worlverhampton, UK for the realization of this work.

References

[1] Mazer, M. (2006) Auditing Databases for Compliance and Risk Management. *DM Review* [online], 16(3), pp. 18 [Accessed 28th April 2013].

[2] Desmond, E. *et al.*, (2007) Oracle Audit Vault Auditor's Guide 10g [online], *Release 2 (10.2.2)* [Accessed 10th May 2013]. Available at: <http://docs.oracle.com/cd/B25369_01/server.102/b28853.pdf>

[3] Aggarwal, V., Aggarwal, N., Kumar, A. and Khatter, H. (2012) Analysis the effect of data mining techniques on database. *Advances in Engineering Software* [online], **47**(1), pp. 164-169 [Accessed 7th June 2013]. Available at:<http://www.sciencedirect.com/science/article/pii/S0965997812000038>.

[4] Johnson, C. and Grandison, T. (2007) Compliance with data protection laws using Hippocratic Database active enforcement and auditing. *IBM Systems Journal* [online], **46**(2), pp. 255-264 Available at:<http://search.proquest.com/docview/222436311 >

[5] Anonymous (2009) Oracle Audit Vault [online]. *An Oracle White Paper* .[Accessed 10th June 2013]. Available at: <http://www.oracle.com/us/products/database/056887.pdf>

[6] Anonymous (2012) Oracle Database Vault Best Practices [online]. *An Oracle White Paper* [Accessed 5th June 2013]. Available at:<http://www.oracle.com/technetwork/database/security/twp-database-vault-bestpractices-132020.pdf>

[7] Kost, S. (2006) Guide to Auditing in Oracle Applications. *white paper* [online]. [Accessed 7th June 2013]. Available at: <http://www.integrigy.com/files/Integrigy_Oracle_11i_Auditing.pdf >

[8] Qing, Z., Sheng, W. and Bin L. (2010) Research and design of fine-grained permissions based on Quality of Detection and Management System [online]. *IEEE*, pp.V3-331-V3-334 [Accessed 12th May 2013]. Available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5608358>

[9] Cheng Z. (2012) *OBJECT-RELATIONAL DATABASE APPROACH FOR ROLE-BASED ACCESS CONTROL (RBAC)* [online]. MSc. Dissertation. USA: California State University, Sacramento. [Accessed 4th May 2013]. Available at: < http://csus-dspace.calstate.edu/bitstream/handle/10211.9/1731/MS_project%20report_Zhimin%20Cheng_Final.pdf?sequence=1>

[10] Li, N.and Mao,Z. (2007) Administration in role-based access control [online]. *ACM*, pp.127-138 [Accessed 16th May, 2013]. Available at: < http://dl.acm.org/citation.cfm?id=1229305>

[11] Murray, M. C. (2010) Database Security: What Students Need to Know. *Journal of Information Technology Education* [online], **9**pp. IIP-61(17) [Accessed 10th May 2013]. Available at: : <http://jite.org/documents/Vol9/JITEv9IIPp061-077Murray804.pdf>.

[12] Huey, P. (2012) Oracle Database Security Guide 11g [online]. *Release 2 (11.2)* .[Accessed 10th May 2013]. Available at: <http://docs.oracle.com/cd/E11882_01/network.112/e16543.pdf>

[13] Carl, D. (2012b) *Database Privileges*. Lecture 7: Database Technology [online]. [Accessed 15 May 2013]. Available at: <http://wolf.wlv.ac.uk/stech/72043/privileges_cp3056.ppt?menu=833937>

[14] Larner, C. (2008) Auditing the DBA: What non-technical managers and auditors should know [online]. *White Paper Absolute Technologies, Inc.* [Accessed 9th June 2013]. Available at: <http://www.absolute-tech.com/download/Auditing%20the%20DBA%20 %20Whitepaper.pdf>

[15] Carl, D. (2012) OKUG: *Auditing techniques For Oracle Database 11g.* UK Oracle User Group 3rd – 5th December. Birmingham: ICC.

[16] Noreen, Z., Hameed, I. and Usman, A. (2009) Development of database auditing infrastructure [online]. *ACM*, pp.1-6 [Accessed 25th April 2013]. Available at: < http://dl.acm.org/citation.cfm?id=1838092>

[17] Bryla, B and Loney K. (2008) *Oracle Database 11g: DBA Handbook*. New York: McGraw-Hill.

[18] Itai, Y., Oludele, A. and Goga, N. (2013) Trigger and Database Security. *International Journal of Computers & Technology* [online], **4**(1), pp. 57-62 [Accessed 7th June 2013]. Available at:< http://cirworld.ijssronline.com/index.php/ijct/article/view/741>

[19] Fabbri, D., Ramamurthy, R. and Kaushik, R. (2013) SELECT Triggers for Data Auditing [online]. *IEEE*, pp.1141-1152 [Accessed 7th July 2013]. Available at: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6544904&tag=1 >.

[20] *Nanda A. (2010) Technology: Security* - Managing Audit Trails. *Oracle Magazine* [online] November, 2010. [Accessed 7th August 2013]. Available at: <http://www.oracle.com/technetwork/issue-archive/2010/10-nov/o60security-176069.html>

[21] Carl, D. (2012a) *Database Auditing*. Lecture 8: Database Technology [online]. [Accessed 15 May 2013]. Available at: <http://wolf.wlv.ac.uk/stech/72043/audit_cp3056.ppt?menu=833936>.

[22] Bature U. (2013) Interviewed by S. Isa for *Shirin Yamma* [Radio]. BBC Hausa Radio, 17 July 2013.

Mr Abdulrahman Yusuf is a lecturer II in the Department of Computer Science, Yobe State University (YSU) Damaturu, Nigeria. Mr Yusuf has joined the University in July 2011. Yusuf graduated from the University of Wolverhampton, UK and the Bayero University Kano, Nigeria for M.Sc. and B.Sc. in Computer Science respectively. Yusuf, also has Diploma in Mathematics and Statistics and Post Graduate Diploma in education. He taught Mathematics and Computer courses to students of some schools in his state origin, Yobe, Nigeria such as USCOEGA, TTC Gwio Kura, YUSAD, and JICON. His area of research interest include integrating modern technology in teaching and learning, Database Technology, and Big Data technology. Yusuf published more than five papers, contributed in graduating more than 300 graduates, and secured NITDA MSc. Scholarship which based on performance in 2012.