**The British College**

**KATHMANDU**

**Coursework Submission Coversheet**(individual coursework only)

**Faculty of Arts, Environment and Technology LBU Student Id:**

77178367

**For checking by the student:**

Please ensure all information is complete and correct and attach this form securely to the front of your work before posting it in a coursework collection box.

Award name: BSc (Hons) Computing

Module code: 23269 – 201516

Module name: Advanced Database B

Module run: 2017

Coursework title: 1

Due Date: 28 Jan (2017)

Module leader: (In LBU) Jackie Campbell, Sanela Lazarevski

Module tutor: (In TBC) Dibya Tara Shakya

**TURNITIN** Checked: YES NO ***(please circle)***

Submission date& time: Date: 28 Jan (2017) Time: 10 PM

**Total Word Count: Total Number of Pages (including this front sheet):**

**In submitting this form with your assignment, you make the following declaration:**  
I declare, that the coursework submitted is my own work and has not (either in whole or part) been submitted towards the award of any other qualification either at LBU or elsewhere. I have fully attributed/referenced all sources of information used during the completion of my assignment, and I am aware that failure to do so constitutes an assessment offence.

Signed: Date:

**You are strongly advised to retain a second copy of your work in case of any query about the assignment.**

3433

23

**For completion by the faculty:**

**This mark is provisional and subject to moderation and approval by the relevant examining board**

**Teacher's Feedback**

**Teacher's Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Contents

[Introduction 3](#_Toc473397761)

[Contractor User Control 3](#_Toc473397762)

[Necessity for Feature 3](#_Toc473397763)

[Security Plan 3](#_Toc473397764)

[Implementation Method 3](#_Toc473397765)

[Role 4](#_Toc473397766)

[Grant 4](#_Toc473397767)

[Procedure 4](#_Toc473397768)

[Trigger 4](#_Toc473397769)

[Application Login 4](#_Toc473397770)

[SQL from SQL 4](#_Toc473397771)

[Revoke 4](#_Toc473397772)

[Implementation 5](#_Toc473397773)

[DDL Scripts and Implementation 5](#_Toc473397774)

[Insert user into application user table 6](#_Toc473397775)

[Creating Role 6](#_Toc473397776)

[Procedure to Create Contractor and User 6](#_Toc473397777)

[Procedure to Grant Role to User 8](#_Toc473397778)

[Trigger 8](#_Toc473397779)

[Create application page using procedure 8](#_Toc473397780)

[SQL from SQL 9](#_Toc473397781)

[Allow Contractor to view their own information only 10](#_Toc473397782)

[Auditing Management 11](#_Toc473397783)

[Audit tables 11](#_Toc473397784)

[Trigger for Audit 13](#_Toc473397785)

[Altering Placement Table 14](#_Toc473397786)

[Testing 15](#_Toc473397787)

[Test Cases 15](#_Toc473397788)

[Test Log 16](#_Toc473397789)

[Test Case 1 16](#_Toc473397790)

[Create Application Users 17](#_Toc473397791)

[Test Case 2 18](#_Toc473397792)

[Test Case 3 18](#_Toc473397793)

[Test Case 4 18](#_Toc473397794)

[Test Case 5 19](#_Toc473397795)

[Test case 6 20](#_Toc473397796)

[Test case 7 20](#_Toc473397797)

[Test case 8 20](#_Toc473397798)

[Test Logs 21](#_Toc473397799)

[Summary 22](#_Toc473397800)

# Introduction

**Database security** is essential aspect of any database project and involves range of security controls. Database security includes security of database and its objects such as DB applications, functions, triggers, stored procedures and data itself. PlaceU recruitment agency has database application that stores information about their contractors, consultants and account companies. It is critical to ensure security of database to protect data from unauthorized users or misuse or unintentional access by users.

Security plan for PlaceU recruitment database application identifies and applies various security policies to strengthen the database security. This Overview report of security plan for PlaceU application plans, implements and test implemented security controls through the use of test cases and test logs.

## Contractor User Control

**User control** is important aspect of PlaceU database application as it is their core authentication and authorization function. Contractor needs to be able to log into the system and system must manage the process. Only authorized/registered users would be able to access the system. Contractor user control would also handle the security of data. Contractors would have access to their own data only. This means they would only see their own placement details and their own account company information. And finally, all the changes made by any contractor would be audited and logged. This will help to track the changes in database and improve data security.

# Necessity for Feature

Contractor User Control is fundamental security feature for PlaceU database application. It manages access control, authentication, application security and is core requirement for auditing process. Without proper user control any registered/unregistered user would use the application and have access data. They could damage the system with ease. With proper user control, system would manage authentication and authorization. Database auditing is possible and database administrator would be able to track changes.

# Security Plan

1. Manage user logging process.
2. Allow contractors to view their own data
3. Manage/Audit changes made by contractor

# Implementation Method

Implementing security plan requires to identify various database security feature to plan and implement them as required. This includes identification and planning of roles, grants, procedures etc. and their implementations.

## Role

Role is group of rights that can be assigned to another role or database users. For a user to able to connect to database itself requires proper role assigned to them. A role with connect, create session and resource privilege is assigned to each user so that they would be able to connect to the system.

## Grant

Grant is used for providing privileges to roles or users. Planned roles for users in this security plan would be assigned to users and roles using grant statement.

## Procedure

To implement this security plan for user control requires a procedure that will be used for creating new contractor as well as creating database user and store linking information between user and contractor in an application table.

## Trigger

Triggers in this security plan is utilized in many occasions. Storing created by and created date is manage by implementing triggers and system auditing too would be managed via use of triggers.

## Application Login

While developing apex application for PlaceU recruitment agency, only database users are granted access to the application. This means any unregistered users/ apex application users will not be able to use the system.

## SQL from SQL

SQL from SQL is methodology to generate dynamic SQL script. This method is very useful to generate scripts for grant, backup scripts for tables or scripts for altering a table dynamically. In current security feature SQL from SQL is used for generating scripts to grant select privileges on all application tables to registered users.

## Revoke

Revoke security feature is utilized in this security plan to undo the privileged granted to users or roles. This is useful when a contractor account is closed and that any request from that user should be prevented.

# Implementation

PlaceU has provided scripts required for creating tables for database and has also provided data for the system. However, there is need of some changes in the provided scripts before executing them. For example, for auditing placement table now has additional four columns to store created date, created by, updated date and updated by. Similarly, additional table PLACEUSER is will be created to maintain link between database users and contractors. This table would store information of database user and id of their corresponding contractor. PlaceU consultants would use PlaceUadmin apex application to enter and create consultants whereas contractors users would use PlaceUClient application to login, view and modify their information.

## DDL Scripts and Implementation

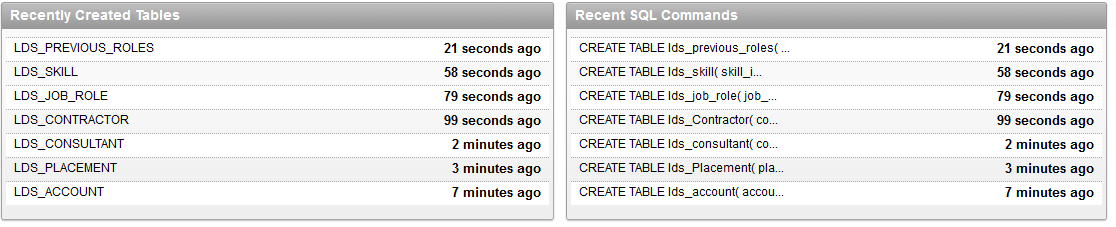


Figure Tables required for database application is created.

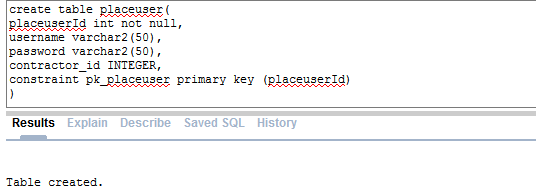


Figure PLACEUSER table is created

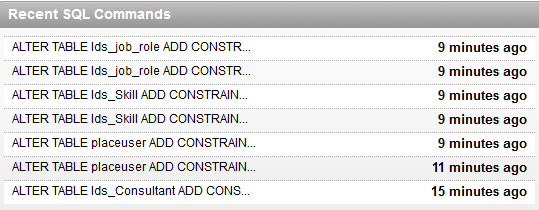


Figure Relation between tables is maintained

## Insert user into application user table

As mention above in this overview document, database user and application link information is stored in PLACEUSER table, CONTRATORPROJECT user information is stored in PLACEUSER table.

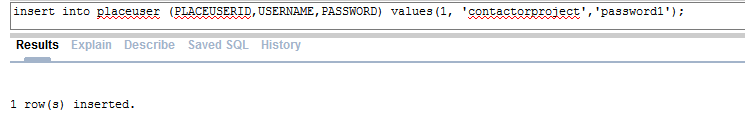


Figure Inserting CONTRACTORPROJECT user info in application user table

# Creating Role

All contractor user will be able to log into the system. To do so certain privileges are required to be granted to users. A new roles is created to with these privileges.

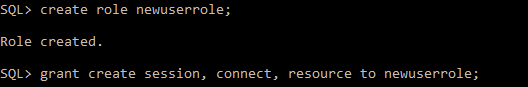


Figure New Role with connect, create session and resource privilege.

# Procedure to Create Contractor and User

Contractor are given a form to register with PlaceU recruitment agency. Information in form is then used by admin to create contractor in PlaceU database application. Here a procedure is executed to perform contractor creation job as well as database user creation for contractor and link with application user table.

create or replace Procedure InsertUser(CONTRACTOR\_Id IN Number,CON\_NAMe in Varchar2,CON\_POSTCODe IN varchar2, CON\_SKILl\_1 IN NUMBER,CON\_SKILl\_2 IN

NUMBER,CON\_SKILl\_3 IN

NUMBER, HIGHEST\_QUAl IN NUMBER, PREFERRED\_ROLe IN Number

)

IS

var\_assign\_user varchar2(1000);

v\_id number;

BEGIN

insert into lds\_contractor(CONTRACTOR\_ID,CON\_NAME,CON\_POSTCODE, CON\_SKILL\_1,CON\_SKILL\_2,CON\_SKILL\_3, HIGHEST\_QUAL, PREFERRED\_ROLE) VALUES

(CONTRACTOR\_Id,CON\_NAMe, CON\_POSTCODe, CON\_SKILl\_1,CON\_SKILl\_2,CON\_SKILl\_3, HIGHEST\_QUAl, PREFERRED\_ROLe)

returning Contractor\_Id into v\_id;

insert into placeuser(username,password,Contractor\_Id) Values (Replace(CON\_NAMe,' ','\_'),Replace(CON\_NAMe,' ','\_'),v\_id);

var\_assign\_user := 'create user ' || Replace(CON\_NAMe,' ','\_') || ' identified by '||Replace(CON\_NAMe,' ','\_');

EXECUTE IMMEDIATE var\_assign\_user;

grantroles(Replace(CON\_NAMe,' ','\_'));

EXCEPTION

WHEN OTHERS THEN

raise\_application\_error(-20001,'An error was encountered - '||SQLCODE||' -ERROR- '||SQLERRM);

END;

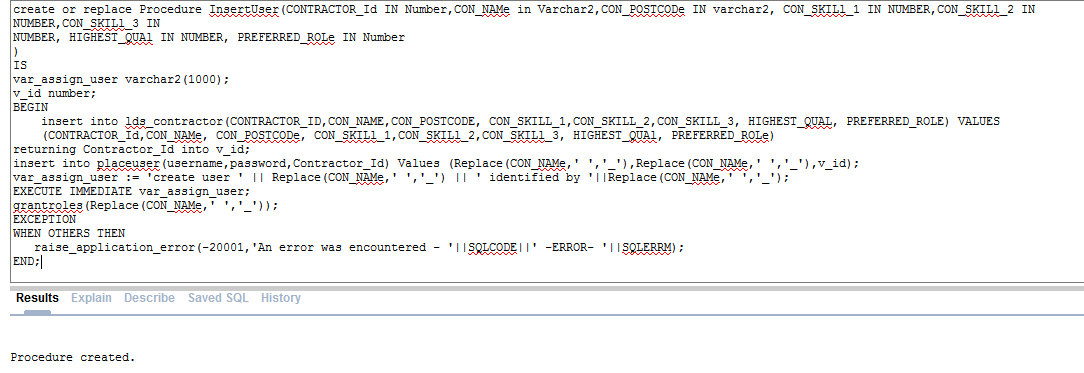


Figure Procedure to create contractor and user for contractor

## Procedure to Grant Role to User

After creating contractor and database user for contractor set of privileges must granted to them in order for them to log into the system. A role has been created already for new users above in this security plan implementation document overview. Granting process is done through procedure. This procedure is executed inside InsertUser procedure after creating of database user.

create or replace procedure grantroles(username in Varchar2)

IS

BEGIN

EXECUTE IMMEDIATE 'GRANT newuserrole to ' || username;

END;

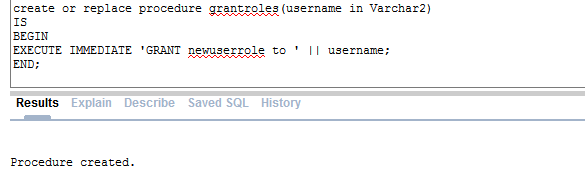


Figure Granting role to Users

## Trigger

A trigger is fired before inserting on LDS\_CONTRACTOR to store created by and created date.

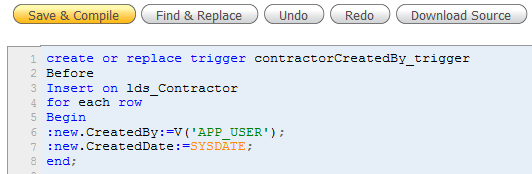


Figure Trigger for created by and created date

## Create application page using procedure

One of the key requirement for executing the created procedure above is to design and develop apex application and allow CONTRACTORPROJECT user to fill up a contractor form. After submitting this form, new contractor would be creating along with database. Necessary information would also be stored in placeuser table and role policy would also be granted to new user.

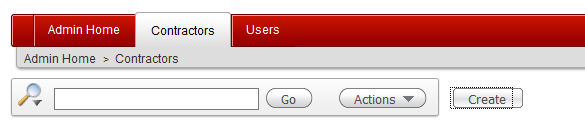


Figure Click on create button

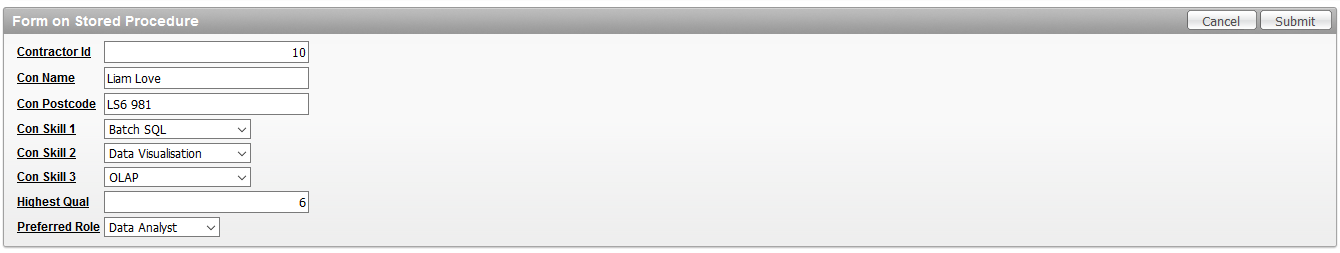


Figure Fill up the form and click submit

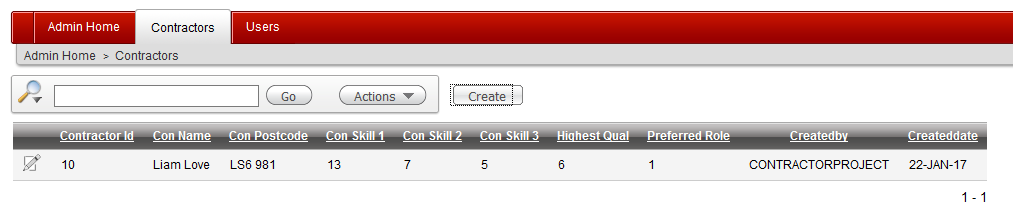


Figure New Contractor is created

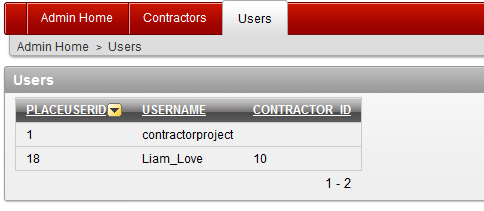


Figure New DB user is created and link with contractor is maintained

## SQL from SQL

Security plan for PlaceU dabatbase application utilizes SQL from SQL feature to generated dynamic scripts to implement security policies. Here scripts for granting select privilege to a user ‘Liam\_Love ’on all tables are generated below.

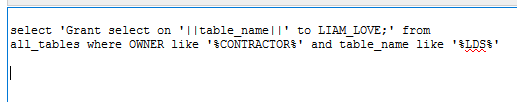


Figure Generating Dynamic SQL using SQL from SQL

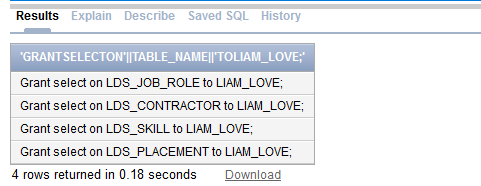


Figure Successful creation of dynamic script using SQL from SQL

# Allow Contractor to view their own information only

One of another key feature of given security plan is to prevent a logged in user from viewing data of another contractor user. This is so much critical for the privacy of a user. Hence a user must be allowed to view and/or modify their own information only. To allow contractors view only their own data only, a where condition is applied when getting the report in apex application. Here in where condition, contractor id of placement table is compared with contractor id of logged in user.

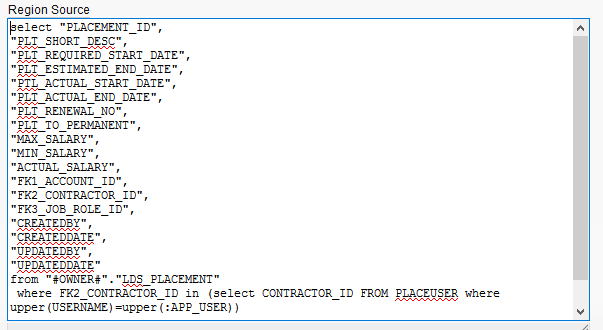


Figure Source of the placement report with where condition to filter own data only

Figure 15 above uses where condition in source which gets the placement report for the user. Any time user logs into the system, user id of logged user can be retrieved from: APP\_USER. As mentioned above to maintain link between application contractor with database user, application table “PLACEUSER” stores contractor\_id and username. This means contractor\_id of logged in user can be taken from PLACEUSER table. Now contractor\_id of logged user is used for filtering displayable data while generating report. This results user gets to view their own data only, securing the privacy of other contractor users.

# Auditing Management

Database auditing is another crucial aspect of database system security. It not only helps in security of database but also helps in maintenance of database. Auditing process includes keeping close eye on user’s activity and keep logs of changes done by users. When a user modifies some information, auditing must be done to ensure back up of original data is stored somewhere in database if the situation arise. In this security plan, any changes in database is audited. If a user modifies some information in a placement information, actions are audited.

For such audit management, two tables are added. One table would store any update action performed in placement table. Another table would be a dedicated audit table for lds\_placement, that would store audit for any information gets deleted.

## Audit tables

First audit table will include columns for storing original(old) value, new value, type of action, data, user, name of table and name of column that got changed.

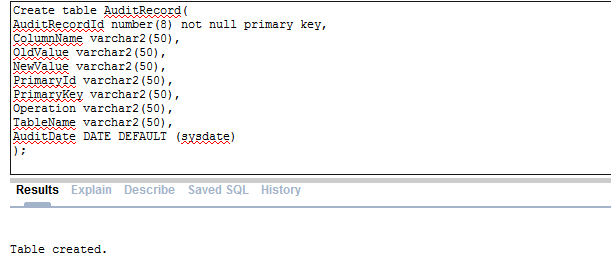


Figure Audit table of LDS\_PLACEMENT

Second audit table would be a mirror table of LDS\_PLACEMENT with some additional columns such as AUDITID to maintain relation with first audit table. This table would be user only if a row in placement table gets deleted. A back up log will be stored in this audit table as back and can be helpful if delete was done mistakenly.

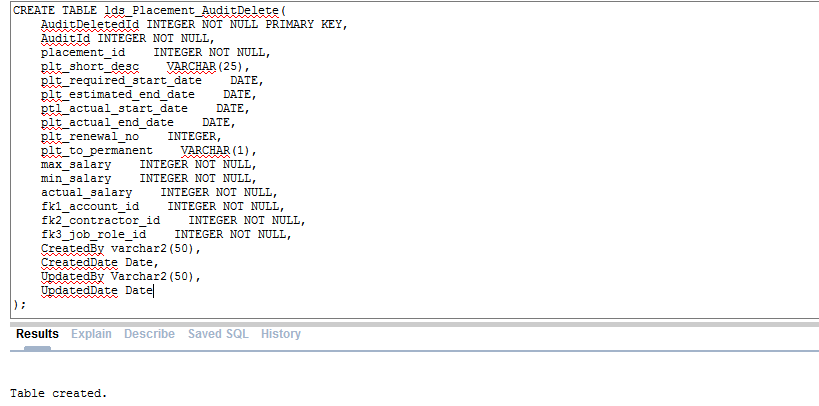


Figure Audit table for Displacement to audit delete actions

Meanwhile sequence and triggers required for primary key sequence for both audit tables are as below.

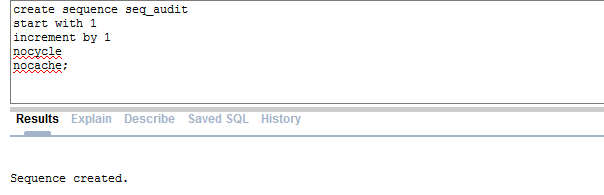


Figure Sequence for audit table

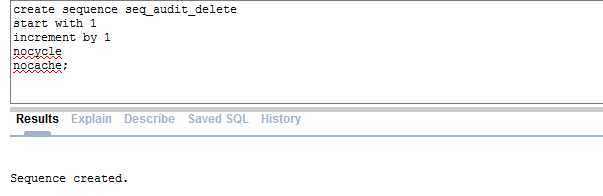


Figure sequence for audit table primary key auto increment

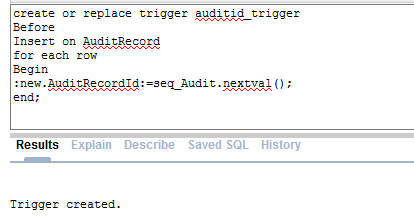


Figure Trigger for incrementing primary key automatically for audit delete table

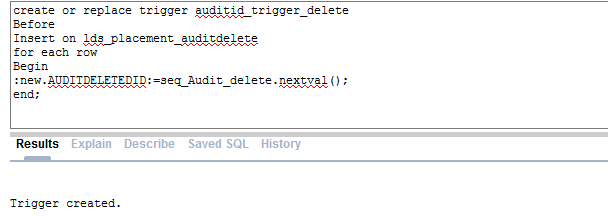


Figure trigger for audit table primary key auto increment

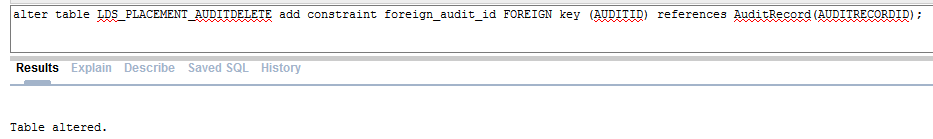


Figure maintaining relation between audit table and audit table for delete

## Trigger for Audit

Along with this paper SQL file is submitted as well which contains scripts for trigger that would audit the update or deletes done in LDS\_Placement. Figure 23 belows is snapshot of trigger being created. When an update or delete action is performed in the placement table, time, user, table, column, old and new value all are audited. However, if delete action is performed, whole row that is deleted is first mirrored in audit delete table.

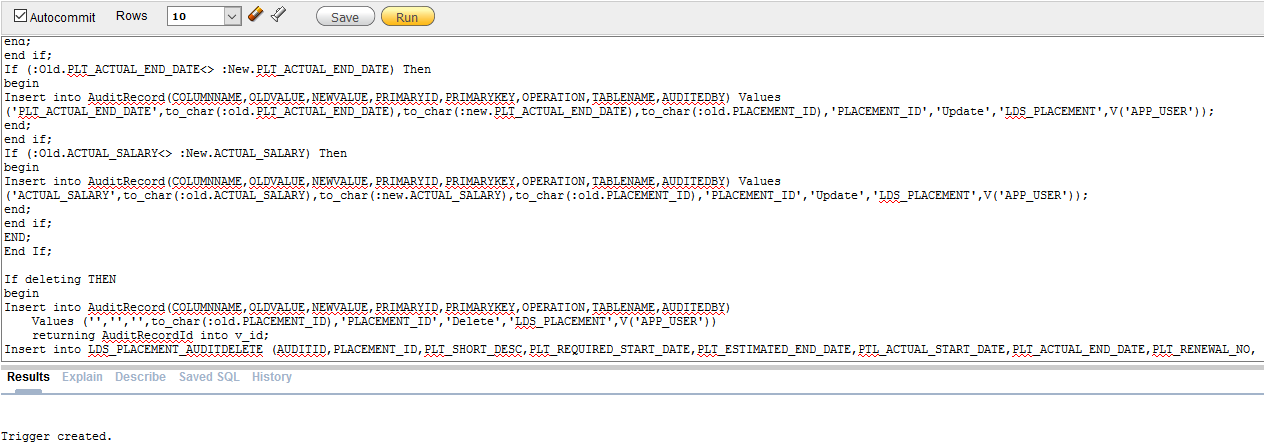


Figure Trigger for audit management

# Altering Placement Table

Another method to keep auditing of a table is to take basic user action and store them in table. For this security plan, created by, created date, updated by and updated date all are stored in table itself as well. This would help to get basic audit report from table itself without need to getting into audit tables.

To perform auditing of changes such as delete or update, provide DLL script for placement table is not appropriate. Hence table should be modified accordingly. Here four rows are added using alternate table statement. These row would store created by, created date, updated by and updated date information.

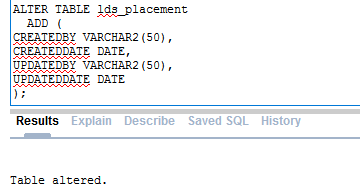


Figure Altering the placement table to store required audit information

Now as the table have required columns, a trigger is created to set updated by and updated date before each update request.

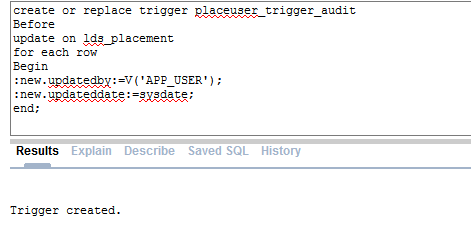


Figure Trigger for auditing update

# Testing

Testing is essential part of database security planning and implementation. Test plan for this database security is managed through test cases and test logs below.

# Test Cases

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Case No.** | **Test Objective** | **Precondition** | **Steps** | **Test** | **Expected Result** | **Post Condition** |
| 1. | Fail Login with Application User | A valid apex application user | Run application🡪 Login Page | Login using application user | Login should Fail. | - |
| 2. | Successful login with database user | A valid database user | Run application🡪 Login Page | Login using valid database user | Login should be successful. | Go to Home page. |
| 3. | Login with user Liam\_Love user should show own data | Liam\_Love User and valid company data | Run Application🡪 Login | Login using Liam\_Love and check companies | Should only show their own company | Go to company List |
| 4. | Login with ContractorProject user should hide Liam\_Love data | ContractorProject | Run Application🡪 Login | Login with ContractorProject and check companies | Should hide Liam\_Love user data | Company List |
| 5. | When update, audit table should keep log | Row in placement table | View placement report🡪 Edit | Update a row | New row in audit table with update log | Placement report |
| 6. | When delete, audit table should keep log | Row in placement table | View placement🡪 Edit | Delete a row | New row in audit table with delete log and mirror data | Placement report |
| 7. | Updating should store updated by and update date | Row in placement for update | View Placement🡪 edit | Update a row | Update by and update date column should be filled | Placement report |
| 8. | Revoke Role | User with newuserole | Run script to revoke newuserrole | Revoke role with create session from user | Connection with user should not be allowed |  |

# Test Log

## Test Case 1

To perform test case 1, first an application user is required. Hence, an application user with username ‘TESTAPEXUSER’ has been created below.

## Create Application Users

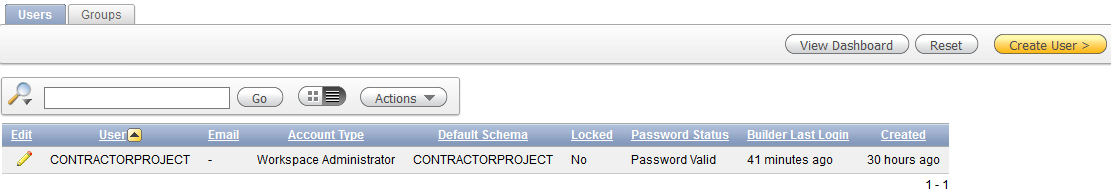


Figure Click on Create in Administration🡪 Users

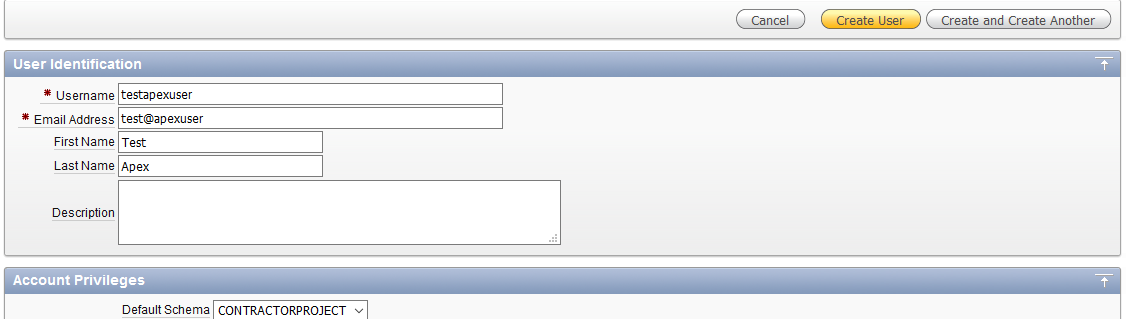


Figure Fill up the form and create user by pressing create user button

Now, as the apex application user has been created, this new user is used for testing the database application. As it’s an apex application user only, it must not be able to log into the system.

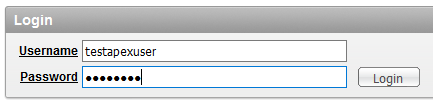


Figure Trying to login using apex user

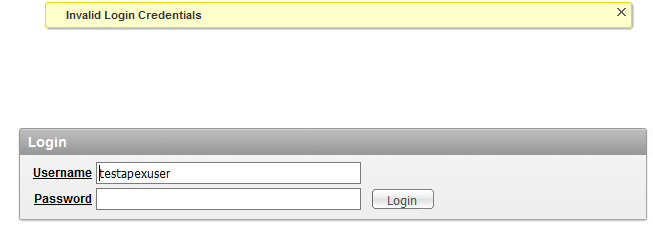


Figure Login Failed when tried to login using apex user

When tried to login using apex application user, system forbidden the user from log in as shown in figure 30 above..

## Test Case 2

In test case 2 database user is used for login into the system for testing. System should allow database user to login to the system.

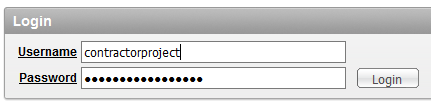


Figure Login using database user



Figure Successful login using database user

Now as shown if figure 32 above, database user could successfully log into the system.

## Test Case 3



Figure login with Liam Love user shows their own data only

Test case 3 shows while login into the system using LIAM\_LOVE user, only their own data are viewable.

## Test Case 4

In this test, another user used for login the system and test if it can view data of user LIAM\_LOVE. To test this, database user CONTRACTORPROJECT logs into the system and check company tab.



Figure Login with CONTRACTORPROJECT user hides data of LIAM\_LOVE user

## Test Case 5

For this test, first in view placements menu, edit button of row to be edited is clicked as shown in figure 34.

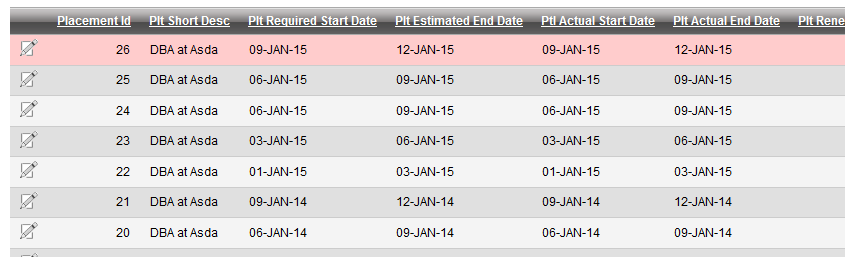


Figure Placement list

Now make changes to the form and apply changes as shown in figure 35.

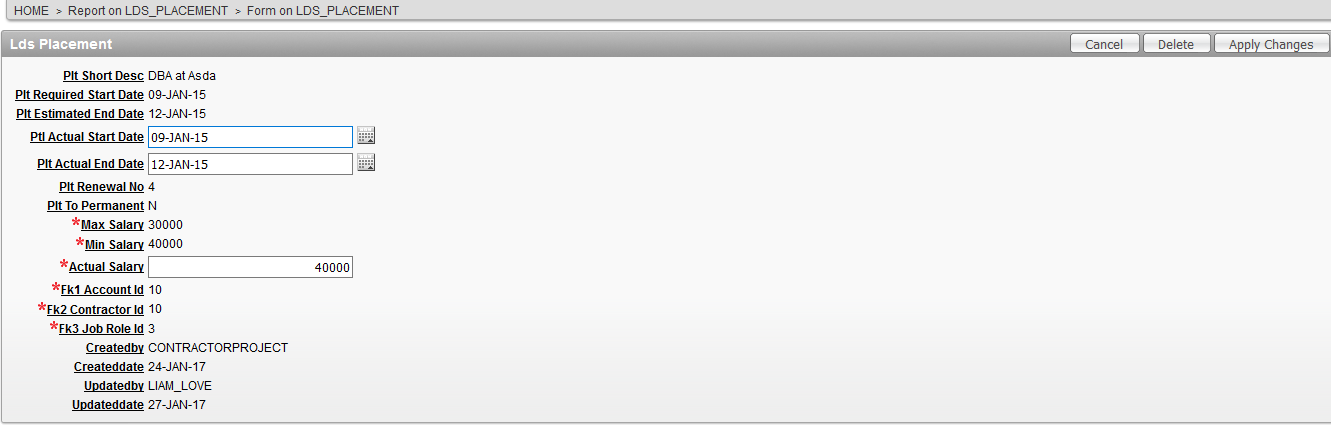


Figure Edit Form of Placement

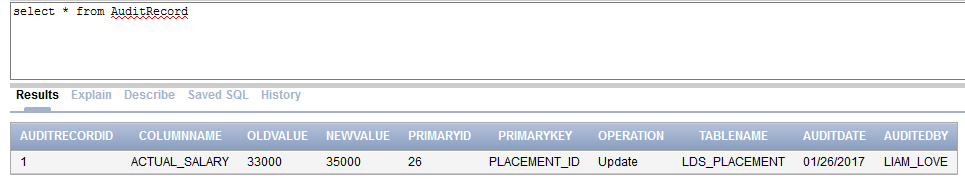


Figure using select statement in audit table shows new audit row

## Test case 6

Now update audit has been tested, deleted test should also be tested. For this a placement information is deleted in figure 37 below.

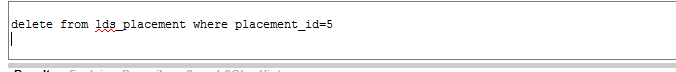


Figure Deleting a row from placement table

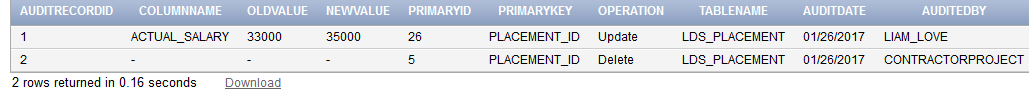


Figure Audit log of deleted action in audit table

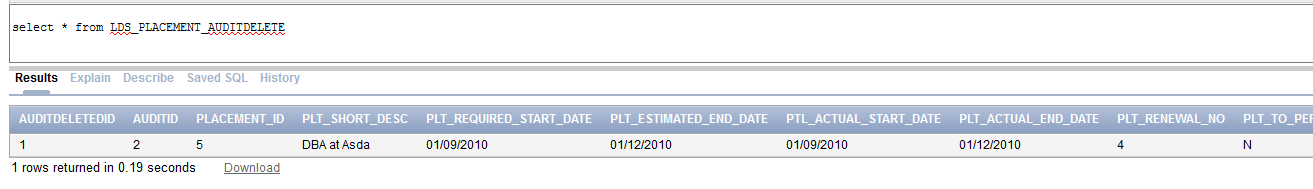


Figure Audit log of deleted table in audit delete table

As shown in figure 38 and 39 above, deleted data is audited in both audit and delete audit tables successfully.

## Test case 7

Test case 7 tests effect of update actions should be audited in original table itself. Select statement is used to check row updated in test case 5 below in figure 40.

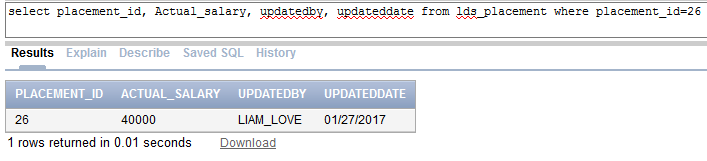


Figure Original table also audits update action

## Test case 8

In this this, user Liam\_love with role newuserorole is used. Newuserrole from the user is revoked and then used for login into database. User should not be able to log into the system as it does not have create session privilege assigned as it was assigned to user via newuserrole.

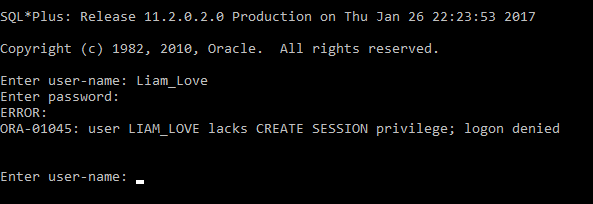


Figure After revoking role, user was not able to login into database system

# Test Logs

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Expected Result | Actual Result | Date |
| 1 | Login should fail when using apex application user | Login failed | Jan 22 2017 |
| 2 | Login should success when using database user | Successful login | Jan 22 2017 |
| 3. | Login with Liam\_Love user should display his own data | User’s own data was shown. | Jan 28 2017 |
| 4. | Login with contractorproject user should hide liam\_love’s user data | Liam\_love’s data was hidden from contractorproject user | Jan 28 2017 |
| 5. | Updating action should be logged into audit table | Update audit is logged | Jan 28 2017 |
| 6. | Deleting action should be logged into both audit and audit delete table | Delete action is logged into both audit tables | Jan 28 2017 |
| 7. | Update action should be audited into original table as well. | Updated action was audited in original table as well. | Jan 28 2017 |
| 8. | Revoking privilege from user should disable user from creating session | User was unable to create session while trying to connect | Jan 28 2017 |

# Summary

This document overview for security planning and implementation of PlaceU database application reports various security policies of plan, implementation procedure and testing of the implanted policies. Various database features such as grant, triggers, procedure, revoke etc. are utilized to implement the planed database security. After implementation and testing, User control for this database application has been managed and any unauthorized user can be prevented from accessing the system. However, system should be given additional level of security from physical damages. Every consultants and contractor should be aware of important of database application security.