University Research Library

Introduction:

This project will be about university research library, not regular research but also sensitive ones. In this project, there are multi-access levels for all the users at the university. Where a regular student could be classified as Unclassified and he can read . Then we can see researcher student user who can read a letter pet more sensitive research and could be classified as Confidential. Then we can see professor user how can deal with more secret research and will be classified as Secret. At the top, we can see the manager of all users could be classified as Top_Secret. In addition, it shows only the articles which are suitable for the user.

Timeline:

- Some series of steps that the project followed in order.
- Choosing a topic (one day).
- write a small description (one day).
- Define functions and policies (two days).
- Project theoretical draft (two days).
- More reading on the required techniques (two days).
- Designing (two days).
- Implementation (4 days).
- Add security policy (two days).
- Testing (one day).
- Reviewing (one day).
- Delivering the project.

Conceptual Model:

Users:

- Student: student user can search research but with the lowest security tag and he can only update his own address.

- Researcher student: researcher student user can search for research from all other students. also, update his own information his adress.

- Professor: professor user can search for all research come from students and other Professors. In addition, he can update his own information.

- Library manager: can select, update, insert, and delete any research and any user. Also, he can check the all tables and modify it.

Constraints:

- The users cannot see research above their domain.
- The users can update thier only adresses.

Entity types and relationships:

We have two entities in this project which are the user entity and the research entity the relation between them that research entity belongs to a user entity which is one to one relation.

Logical Model:

The users have SSN as Identifier
And their attributes are:

- name

- major

- user type (student ,research student , professor,)

- Phone number.

For sensitive Articles table there is Article_ID as Identifier
Also there are:

-The Author name

-body of the article

-And date of the article.

_____
Implementation and Physical Model:
_____

```
SQL> Connect as SYSDBA;
Enter user-name: mandub
Enter password:
Connected.

### Reminder: To apply OLS  we must complete 2 steps to prepare.
# First, we must create a pluggable database ( a pdb).
# Second, we must configure our pdb (that we just created).

SQL> ALTER SESSION SET CONTAINER=OLS ;
Session altered.

SQL>  SHOW CON_NAME;
CON_NAME
------------------------------
OLS

SQL> SELECT PDB_NAME FROM DBA_PDBS;
SELECT PDB_NAME FROM DBA_PDBS
                     *
ERROR at line 1:
ORA-01219: database or pluggable database not open: queries allowed on fixed
tables or views only

SQL> startup;
Pluggable Database opened.

SQL> SELECT PDB_NAME FROM DBA_PDBS;
PDB_NAME
--------------------------------------------------------------------------------
OLS

SQL> SELECT NAME, OPEN_MODE FROM V$PDBS;
NAME                           OPEN_MODE
------------------------------ ----------
OLS                            READ WRITE
```

```
SQL> SELECT NAME, STATUS, DESCRIPTION FROM DBA_OLS_STATUS;
NAME                 STATU
-------------------- -----
DESCRIPTION
--------------------------------------------------------------------------------
OLS_CONFIGURE_STATUS TRUE
Determines if OLS is configured

OLS_DIRECTORY_STATUS FALSE
Determines if OID is enabled with OLS

OLS_ENABLE_STATUS    TRUE
Determines if OLS is enabled

SQL> EXECUTE LBACSYS.CONFIGURE_OLS;
PL/SQL procedure successfully completed.

SQL> EXECUTE LBACSYS.OLS_ENFORCEMENT.ENABLE_OLS;
PL/SQL procedure successfully completed.

SQL> GRANT INHERIT PRIVILEGES ON USER SYS TO LBACSYS;
Grant succeeded.

SQL> GRANT LBAC_dba TO SYS;
Grant succeeded.

SQL> CONNECT SYS AS SYSOPER;
Enter password:
Connected.
SQL> show user;
USER is "PUBLIC"

SQL> Connect as SYSDBA;
Enter user-name: mandub
Enter password:
Connected.

SQL> alter session set container=OLS;
Session altered.
--------------------------------------------------------------------------------
################################################################################
 # Now we create an OLS policy that will do the following
### a. The policy is named OLSS, with a column_name of SS with READ_CONTROL;

SQL> EXECUTE SA_SYSDBA.CREATE_POLICY ('OLSS','SS', 'READ_CONTROL');
PL/SQL procedure successfully completed.
```

b. The Security Levels in the policy are the hierarchical levels that described as flowing:

Level 100 - U - UNCLASSIFIED
Level 200 - C - CONFIDENTIAL
Level 300 - S - SECRET
Level 400 - TS- TOP_SECRET

```
SQL> EXECUTE SA_COMPONENTS.CREATE_LEVEL ('OLSS', 100, 'U', 'UNCLASSIFIED');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_COMPONENTS.CREATE_LEVEL ('OLSS', 200, 'C', 'CONFIDENTIAL');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_COMPONENTS.CREATE_LEVEL ('OLSS', 300, 'S', 'SECRET');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_COMPONENTS.CREATE_LEVEL ('OLSS', 400, 'TS', 'TOP_SECRET');
PL/SQL procedure successfully completed.
```

---

c. The Data Labels are as required to implement the policy are (Data Labels of U, C, S, and TS);

```
SQL> EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('OLSS', 100, 'U');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('OLSS', 200, 'C');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('OLSS', 300, 'S');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_LABEL_ADMIN.CREATE_LABEL('OLSS', 400, 'TS');
PL/SQL procedure successfully completed.
```

##

Creating users :

We have two users in manager position (manager and manager2).

```
SQL> Create user manager identified by m;
User created.

SQL> GRANT CONNECT TO manager;
Grant succeeded.

SQL> Grant create session to manager;
Grant succeeded.

SQL> GRANT CONNECT, RESOURCE, DBA TO manager;
```

```
Grant succeeded.

SQL> GRANT UNLIMITED TABLESPACE TO manager;
Grant succeeded.

---------------------------
SQL> Create user manager2 identified by m2;
User created.

SQL> GRANT CONNECT TO manager2;
Grant succeeded.

SQL> Grant create session to manager2;
Grant succeeded.

SQL> GRANT CONNECT, RESOURCE, DBA TO manager2;
Grant succeeded.

SQL> GRANT UNLIMITED TABLESPACE TO manager2;
Grant succeeded.

################################################################################
# Creating needed tables { All_Users , Articles}

# Table All_Users table hold user informations.
# Users can only update some of their adress.

SQL> create table manager.All_Users (SSN number (4) not null,
  2  NAME VARchar(10),
  3  Major VARchar(10),
  4  Address VARchar(16),
  5  user_type VARchar(16),
  6  Phone NUMBER(8));
Table created.
##############


# Articles will be protected by OLD. The access to this table should be
# limited by the users level.

SQL> create table manager.Articles (Article_ID number (4) not null,
  2  author_name VARchar(20),
  3  Title  VARchar(20),
  4  body  VARchar(40),
  5  art_date VARchar(20));
Table created.
################################################################################
```

```
# Grant user manager2 all needed access to the tables
SQL> Grant select,update,delete,insert on manager.All_Users to manager2;
Grant succeeded.

SQL> Grant select,update,delete,insert on manager.Articles to manager2;
Grant succeeded.

################################################################################

# continue creating users:
Grant select on Articles the users for OLS demonstration reason

SQL> Create user Student1 identified by S1;
User created.

SQL> Grant create session to Student1;
Grant succeeded.

SQL> Grant select on manager.Articles to Student1;
Grant succeeded.
#################

SQL> Create user Student2 identified by S2;
User created.

SQL> Grant create session to Student2;
Grant succeeded.

SQL> Grant select on manager.Articles to Student2;
Grant succeeded.
#################
SQL> Create user St_res1 identified by SR1;
User created.

SQL> Grant create session to St_res1;
Grant succeeded.

SQL> Grant select on manager.Articles to St_res1;
Grant succeeded.
#################

SQL> Create user St_res2 identified by SR2;
User created.

SQL> Grant create session to St_res2;
Grant succeeded.

SQL> Grant select on manager.Articles to St_res2;
```

```
Grant succeeded.
################

SQL> Create user professor1 identified by P1;
User created.

SQL> Grant create session to professor1;
Grant succeeded.

SQL> Grant select on manager.Articles to professor1;
Grant succeeded.
###############

SQL> Create user professor2 identified by P2;
User created.

SQL> Grant create session to professor2;
Grant succeeded.

SQL> Grant select on manager.Articles to professor2;
Grant succeeded.
```

```
### d. That the Users have the MAC rights as indicated in the project description
        The Level for Student1 and  Student2 is U.
        The Level for St_res1 and St_res2    is C.
        The Level for professor and professor2 is S.
        The Level for manager2                is TS.

SQL>  alter session set container=OLS;
Session altered.

SQL> EXECUTE SA_USER_ADMIN.SET_LEVELS ('OLSS', 'Student1', 'U', 'U', 'U', 'U');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_LEVELS ('OLSS', 'Student2', 'U', 'U', 'U', 'U');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_LEVELS ('OLSS', 'St_res1', 'C', 'U', 'C', 'C');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_LEVELS ('OLSS', 'St_res2', 'C', 'U', 'C', 'C');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_LEVELS ('OLSS', 'professor1', 'S', 'U', 'S', 'S');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_LEVELS ('OLSS', 'professor2', 'S', 'U', 'S', 'S');
```

```
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_LEVELS ('OLSS', 'manager2', 'TS', 'U', 'TS', 'TS');
PL/SQL procedure successfully completed.


################################################################################
SQL> SELECT * FROM DBA_SA_USER_LEVELS;
POLICY_NAME
------------------------------
USER_NAME
--------------------------------------------------------------------------------
MAX_LEVEL                      MIN_LEVEL
------------------------------ ------------------------------
DEF_LEVEL                      ROW_LEVEL
------------------------------ ------------------------------
OLSS
MANAGER2
TS                             U
TS                             TS

POLICY_NAME
------------------------------
USER_NAME
--------------------------------------------------------------------------------
MAX_LEVEL                      MIN_LEVEL
------------------------------ ------------------------------
DEF_LEVEL                      ROW_LEVEL
------------------------------ ------------------------------
OLSS
ST_RES2
C                              U
C                              C


POLICY_NAME
------------------------------
USER_NAME
--------------------------------------------------------------------------------
MAX_LEVEL                      MIN_LEVEL
------------------------------ ------------------------------
DEF_LEVEL                      ROW_LEVEL
------------------------------ ------------------------------
OLSS
STUDENT1
U                              U
U                              U
```

```
POLICY_NAME
------------------------------
USER_NAME
--------------------------------------------------------------------------------
MAX_LEVEL                      MIN_LEVEL
------------------------------ ------------------------------
DEF_LEVEL                      ROW_LEVEL
------------------------------ ------------------------------
OLSS
STUDENT2
U                              U
U                              U

POLICY_NAME
------------------------------
USER_NAME
--------------------------------------------------------------------------------
MAX_LEVEL                      MIN_LEVEL
------------------------------ ------------------------------
DEF_LEVEL                      ROW_LEVEL
------------------------------ ------------------------------
OLSS
PROFESSOR2
S                              U
S                              S


POLICY_NAME
------------------------------
USER_NAME
--------------------------------------------------------------------------------
MAX_LEVEL                      MIN_LEVEL
------------------------------ ------------------------------
DEF_LEVEL                      ROW_LEVEL
------------------------------ ------------------------------
OLSS
PROFESSOR1
S                              U
S                              S


POLICY_NAME
------------------------------
USER_NAME
--------------------------------------------------------------------------------
MAX_LEVEL                      MIN_LEVEL
------------------------------ ------------------------------
```

```
DEF_LEVEL                      ROW_LEVEL
------------------------------ ------------------------------
OLSS
ST_RES1
C                              U
C                              C
################################################################################

SQL> SELECT USER_NAME FROM DBA_SA_USERS;

USER_NAME
--------------------------------------------------------------------------------
MANAGER2
STUDENT1
STUDENT2
ST_RES1
ST_RES2
PROFESSOR1
PROFESSOR2
```

---

e. That the OLS policy created is applied to the manager.Articles table.

```
SQL> EXECUTE SA_POLICY_ADMIN.APPLY_TABLE_POLICY ('OLSS', 'manager', 'Articles');
PL/SQL procedure successfully completed.

######## INSERT vlues to manager.Articles table

SQL> INSERT INTO manager.Articles VALUES (1, 'ADAMS','Physics',
  2  'physics general theory','1-1-2019',100);
1 row created.

SQL> INSERT INTO manager.Articles VALUES (2, 'BAKER','nuclear research',
  2  'nuclear general theory','1-2-2019',200);
1 row created.


SQL> INSERT INTO manager.Articles VALUES (3, 'CHUCK','secret research',
  2  'nuclear secret research','1-3-2019',300);
1 row created.

SQL> INSERT INTO manager.Articles VALUES (4, 'DONNER','top secret',
  2  'nuclear top secret research','1-4-2019',400);
1 row created.

SQL> SELECT * FROM manager.Articles;

ARTICLE_ID AUTHOR_NAME          TITLE
---------- -------------------- --------------------
```

```
BODY                                     ART_DATE                     SS
---------------------------------------- -------------------- ----------
         1 ADAMS                Physics
physics general theory                   1-1-2019                    100

         2 BAKER                nuclear research
nuclear general theory                   1-2-2019                    200

         3 CHUCK                secret research
nuclear secret research                  1-3-2019                    300


ARTICLE_ID AUTHOR_NAME          TITLE
---------- -------------------- --------------------
BODY                                     ART_DATE                     SS
---------------------------------------- -------------------- ----------
         4 DONNER               top secret
nuclear top secret research              1-4-2019                    400
```

```
################################################################################
# Set users privileges.

SQL> EXECUTE SA_USER_ADMIN.SET_USER_PRIVS ('OLSS', 'Student1', 'READ');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_USER_PRIVS ('OLSS', 'Student2', 'READ');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_USER_PRIVS ('OLSS', 'St_res1', 'READ');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_USER_PRIVS ('OLSS', 'St_res2', 'READ');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_USER_PRIVS ('OLSS', 'professor1', 'READ');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_USER_PRIVS ('OLSS', 'professor2', 'READ');
PL/SQL procedure successfully completed.

SQL> EXECUTE SA_USER_ADMIN.SET_USER_PRIVS ('OLSS', 'manager2', 'FULL');
PL/SQL procedure successfully completed.


_____________________________________
SQL> SELECT USER_NAME,USER_PRIVILEGES FROM DBA_SA_USERS;

USER_NAME
--------------------------------------------------------------------------------
```

```
USER_PRIVILEGES
----------------------------------------------------------------------------
STUDENT1
READ


USER_NAME
----------------------------------------------------------------------------
USER_PRIVILEGES
----------------------------------------------------------------------------
STUDENT2
READ

ST_RES1
READ

ST_RES2
READ


USER_NAME
----------------------------------------------------------------------------
USER_PRIVILEGES
----------------------------------------------------------------------------
PROFESSOR1
READ

PROFESSOR2
READ

MANAGER2
FULL

################################################################################
################################################################################

# Demonstration OLS.

Connect STUDENT1/S1;
Connected.
SQL> SELECT * FROM manager.Articles;
ARTICLE_ID AUTHOR_NAME          TITLE
---------- -------------------- --------------------
BODY                                     ART_DATE                     SS
---------------------------------------- -------------------- ----------
         1 ADAMS                Physics
physics general theory                   1-1-2019                    100
```

```
Connect ST_RES1/SR1;
Connected.

SQL> SELECT * FROM manager.Articles;
ARTICLE_ID AUTHOR_NAME          TITLE
---------- -------------------- --------------------
BODY                                     ART_DATE                   SS
---------------------------------------- -------------------- ----------
         1 ADAMS                Physics
physics general theory                   1-1-2019                  100

         2 BAKER                nuclear research
nuclear general theory                   1-2-2019                  200
```

```
Connect PROFESSOR1/P1;
Connected.
SQL> SELECT * FROM manager.Articles;
ARTICLE_ID AUTHOR_NAME          TITLE
---------- -------------------- --------------------
BODY                                     ART_DATE                   SS
---------------------------------------- -------------------- ----------
         1 ADAMS                Physics
physics general theory                   1-1-2019                  100

         2 BAKER                nuclear research
nuclear general theory                   1-2-2019                  200

         3 CHUCK                secret research
nuclear secret research                  1-3-2019                  300
```

```
Connect MANAGER2/m2;
Connected.

SQL> INSERT INTO manager.Articles VALUES (5, 'DONNER','top secret2',
  2  'nuclear top secret research2','1-5-2019',400);
1 row created.

SQL> SELECT * FROM manager.Articles;
ARTICLE_ID AUTHOR_NAME          TITLE
---------- -------------------- --------------------
BODY                                     ART_DATE                   SS
---------------------------------------- -------------------- ----------
         5 DONNER               top secret2
nuclear top secret research2             1-5-2019                  400

         1 ADAMS                Physics
```

```
physics general theory                  1-1-2019                    100

         2 BAKER                nuclear research
nuclear general theory                  1-2-2019                    200


ARTICLE_ID AUTHOR_NAME          TITLE
---------- -------------------- --------------------
BODY                                    ART_DATE                     SS
---------------------------------------- -------------------- ----------
         3 CHUCK                secret research
nuclear secret research                 1-3-2019                    300

         4 DONNER               top secret
nuclear top secret research             1-4-2019                    400
```

```
################################################################################
# Fill in all_users table

SQL> INSERT INTO manager.All_Users VALUES
  2  (1, 'Student1','physic','111 coloma st','st',12345678);
1 row created.

SQL> INSERT INTO manager.All_Users VALUES
  2  (2, 'Student2','physic2','222 mam st','st',22222222);
1 row created.

SQL> INSERT INTO manager.All_Users VALUES
  2  (3, 'St_res1','nuclear1','333 mad st','st_re',33333333);
1 row created.

SQL> INSERT INTO manager.All_Users VALUES
  2  (4, 'St_res2','nuclear2','444 od st','st_re',44444444);
1 row created.

SQL> INSERT INTO manager.All_Users VALUES
  2  (5, 'professor1','nuclear','555 ax st','pro',55555555);
1 row created.

SQL> INSERT INTO manager.All_Users VALUES
  2  (6, 'professor2','nuclear','666 xx st','pro',66666666);
1 row created.

SQL> SELECT * FROM manager.All_Users;

       SSN NAME       MAJOR      ADDRESS          USER_TYPE              PHONE
---------- ---------- ---------- ---------------- ---------------- ----------
         1 Student1   physic     111 coloma st    st                 12345678
```

```
         2 Student2   physic2    222 mam st       st                22222222
         3 St_res1    nuclear1   333 mad st       st_re             33333333
         4 St_res2    nuclear2   444 od st        st_re             44444444
         5 professor1 nuclear    555 ax st        pro               55555555
         6 professor2 nuclear    666 xx st        pro               66666666

6 rows selected.
################################################################################

SQL> Create Role Users_role;
Role created.

SQL> Grant Users_role To Student1,Student2 ,St_res1,St_res2,professor1,professor2;
Grant succeeded.

SQL> Create View User_Update_Address As Select * from All_Users Where Name = User;
View created.

SQL> Grant SELECT on User_Update_Address to Users_role;
Grant succeeded.

SQL> Grant UPDATE(Address ) on User_Update_Address to Users_role;
Grant succeeded.

##  Demonstrate
####################
Connect Student1 /S1;
Connected.

SQL> Select * from manager.User_Update_Address;
       SSN NAME       MAJOR      ADDRESS          USER_TYPE              PHONE
---------- ---------- ---------- ---------------- ---------------- ----------
         1 Student1   physic     111 coloma st    st                 12345678


SQL> Update manager.User_Update_Address Set Address = 'NEW_ADDRESS'
Where Name = User;
1 row updated.


SQL> Update manager.User_Update_Address Set ADDRESS= 'NEW_ADDRESS2'
Where Name = 'Student2';
0 rows updated.


SQL> Select * from manager.User_Update_Address;

       SSN NAME       MAJOR      ADDRESS          USER_TYPE              PHONE
```

```
---------- ---------- ---------- ---------------- ---------------- ----------
         1 Student1   physic     NEW_ADDRESS      st                 12345678
```

```
SQL> Connect SYS/W as SYSDBA;
Connected.
SQL> SELECT * FROM manager.All_Users ;

       SSN NAME       MAJOR      ADDRESS          USER_TYPE             PHONE
---------- ---------- ---------- ---------------- ---------------- ----------
         1 Student1   physic     NEW_ADDRESS      st                 12345678
         2 Student2   physic2    222 mam st       st                 22222222
         3 St_res1    nuclear1   333 mad st       st_re              33333333
         4 St_res2    nuclear2   444 od st        st_re              44444444
         5 professor1 nuclear    555 ax st        pro                55555555
         6 professor2 nuclear    666 xx st        pro                66666666

6 rows selected.
```

##

_____

###Security:

_____

What is your security plan?

My security plan was to protect the table all user by letting user access only their record and only can update their address. In addition, my security plan was to protect table articles by OLS where the user can see only articles from their level or less.

What are your security policies that support that plan?

First, users cannot see other information.

Second, they can update their own address only.

Third, they can access articles that in their level only.

What security procedures will support the plan and policies?

I use the role and view together. Also, I used OLS to apply my plan.

What is your security model?

I have two part one when the users need to change their address.

They have only one row they can see also the have only one filed that they can change.

The second part , I use OLS to help me with the access controls which based on the classification of the data I have four classes (levels) which considered very powerful capability enables access to sensitive data.

And the security implementation

Demonstrated well in implementation part above