

CST8276 Lab 4: User Security

Purpose: This lab is the first of two that address the larger theme of creating users and understanding privileges and roles. These labs are important because they are an example of a role-based access control (RBAC) security paradigm.

The main idea behind this approach is that data access is controlled by assigning predefined named privileges into named roles that are then assigned to a user. The advantage of this approach is that it reduces the complexity of managing resources - and managing them very precisely - especially as the number of resources being guarded and the number of users with disparate or overlapping requirements increases.

Deliverable: To earn 2 marks towards your lab score, submit the requirements listed below in a copy of **this** single document **and demonstrate the results to your lab professor.**

Requirements:

1. (Fill in the blank). On page 3 of this the following document, [http://profsandhu.com/journals/computer/i94rbac\(org\).pdf](http://profsandhu.com/journals/computer/i94rbac(org).pdf), the authors state that RBAC supports the least *Privilege* security principle.
2. (Fill in the blanks). The information in the *About User Accounts* section of the following Oracle document, <http://docs.oracle.com/database/121/ADMQS/GUID-7FC1D8BE-4BB9-4642-A4CE-29CD2B8A5F23.htm#ADMQS007> , identifies at least 6 things that must be done when **creating a user**. List 5 things:
 - a. Assign a username to the account
 - b. Grant appropriate system privileges and roles to the account
 - c. Give the user account a space usage quota on each tablespace
 - d. Assign a password to the account
 - e. Assign a default tablespace
3. Open a command window as “**run as administrator**”. Then run “`sqlplus / as sysdba`”, which will connect you as SYS in the SYSDBA role.
 - a. As a reminder of how to find the relevant V_\$ or system tables containing information you might want to look at or analyze, try the following, and show your output in the box below.

CST8276 Lab 4: User Security

```
Select Administrator: Command Prompt - sqlplus / as sysdba

SQL> select * from dict where table_name like '%QUOTA%';

TABLE_NAME
-----
COMMENTS
-----
DBA_TS_QUOTAS
Tablespace quotas for all users

USER_TS_QUOTAS
Tablespace quotas for the user

SQL>
```

- b. Try the above and “DESC DBA_TABLESPACES;” show both outputs below.

```
Administrator: Command Prompt - sqlplus / as sysdba
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Fri Jun 11 15:39:06 2021

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

SQL> SELECT * FROM DICT WHERE TABLE_NAME LIKE '%QUOTA%';

TABLE_NAME
-----
COMMENTS
-----
DBA_TS_QUOTAS
Tablespace quotas for all users

USER_TS_QUOTAS
Tablespace quotas for the user

Administrator: Command Prompt - sqlplus / as sysdba
Tablespace quotas for the user

SQL> DESC DBA_TABLESPACES;
Name Null? Type
-----
TABLESPACE_NAME NOT NULL VARCHAR2(30)
BLOCK_SIZE NOT NULL NUMBER
INITIAL_EXTENT NUMBER
NEXT_EXTENT NUMBER
MIN_EXTENTS NOT NULL NUMBER
MAX_EXTENTS NUMBER
MAX_SIZE NUMBER
PCT_INCREASE NUMBER
MIN_EXTLEN NUMBER
STATUS VARCHAR2(9)
CONTENTS VARCHAR2(9)
LOGGING VARCHAR2(9)
FORCE_LOGGING VARCHAR2(3)
EXTENT_MANAGEMENT VARCHAR2(10)
ALLOCATION_TYPE VARCHAR2(9)
PLUGGED_IN VARCHAR2(3)
SEGMENT_SPACE_MANAGEMENT VARCHAR2(6)
DEF_TAB_COMPRESSION VARCHAR2(8)
RETENTION VARCHAR2(11)
BIGFILE VARCHAR2(3)
PREDICATE_EVALUATION VARCHAR2(7)
ENCRYPTED VARCHAR2(3)
COMPRESS_FOR VARCHAR2(30)
DEF_INMEMORY VARCHAR2(8)
DEF_INMEMORY_PRIORITY VARCHAR2(8)
DEF_INMEMORY_DISTRIBUTE VARCHAR2(15)
DEF_INMEMORY_COMPRESSION VARCHAR2(17)
DEF_INMEMORY_DUPLICATE VARCHAR2(13)

SQL>
```

CST8276 Lab 4: User Security

- c. Try “SELECT TABLESPACE_NAME, BYTES, MAX_BYTES, (100*(BYTES/MAX_BYTES)) AS PERCENT FROM DBA_TS_QUOTAS;” **Provide a screenshot of your command and the output on your system**

```
SQL> "SELECT TABLESPACE_NAME, BYTES, MAX_BYTES, (100*(BYTES/MAX_BYTES)) AS PERCENT FROM DBA_TS_QUOTAS;"
SP2-0734: unknown command beginning ""SELECT TA..." - rest of line ignored.
SQL> SELECT TABLESPACE_NAME, BYTES, MAX_BYTES, (100*(BYTES/MAX_BYTES)) AS PERCENT FROM DBA_TS_QUOTAS;

TABLESPACE_NAME          BYTES  MAX_BYTES  PERCENT
-----
SYSAUX                   1441792  104857600    1.375
SYSAUX                      0         -1          0
SYSAUX                      0         -1          0
SYSAUX                   1507328         -1 -150732800
EXAMPLE                  10420224         -1 -1.042E+09
SYSAUX                      0         -1          0

6 rows selected.

SQL>
```

- d. Which tablespace is most utilized (based on %)? SYSAUX
- e. Run the following command on your system:
- ```
select TABLESPACE_NAME, CONTENTS FROM
DBA_TABLESPACES;
```

**Which tablespaces are permanent on your system?**

**SYSTEM, SYSAUX, USERS, EXAMPLE**

4. Unlock the “scott” username by using the “alter user ...” command. Show your work. (By the way, later you will need to know the password is “tiger”). Show your work.

Administrator: Command Prompt - sqlplus / as sysdba

```
SQL> ALTER USER scott IDENTIFIED BY tiger;

User altered.

SQL> _
```

# CST8276 Lab 4: User Security

5. Consider the example below:

```

SQL> column GRANTEE format a20
SQL> column GRANTED_ROLE format a30
SQL> select grantee, granted_role from dba_role_privs where grantee='SYSTEM';

GRANTEE GRANTED_ROLE

SYSTEM AQ_ADMINISTRATOR_ROLE
SYSTEM DBA

```

Then, using the DBA\_SYS\_PRIVS, DBA\_TAB\_PRIVS, and DBA\_ROLE\_PRIVS tables, determine all of the privileges that have been **directly** granted to the “SCOTT” username. Show your work below (you can use multiple queries).

```
SQL> COLUMN GRANTEE FORMAT A20;
SQL> COLUMN PRIVILEGE FORMAT A30;
SQL> SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE='SCOTT';
```

| GRANTEE | PRIVILEGE            |
|---------|----------------------|
| SCOTT   | UNLIMITED TABLESPACE |

## CST8276 Lab 4: User Security

- a. Using the DBA\_SYS\_PRIVS, DBA\_TAB\_PRIVS, and DBA\_ROLE\_PRIVS tables, determine the privileges that have been **directly** granted to the “SYSTEM” and “SYS” usernames. Show your work below (you can use multiple queries). Make sure the counts are visible.

```
SQL> COLUMN GRANTEE FORMAT A20;
SQL> COLUMN PRIVILEGE FORMAT A30;
SQL> SELECT GRANTEE, PRIVILEGE FROM DBA_SYS_PRIVS WHERE GRANTEE='SCOTT';

GRANTEE PRIVILEGE
----- -
SCOTT UNLIMITED TABLESPACE

SQL> SELECT GRANTEE, PRIVILEGE FROM DBA_TAB_PRIVS WHERE GRANTOR='SCOTT';

GRANTEE PRIVILEGE
----- -
PUBLIC INHERIT PRIVILEGES

SQL> SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTEE='SCOTT';

GRANTEE

GRANTED_ROLE

SCOTT
RESOURCE

SCOTT
CONNECT

SQL>
```

6. Identify on which tablespaces the “scott” user has been allocated space. Justify your answer.

```
SQL> SELECT TABLESPACE_NAME FROM DBA_TABLESPACES WHERE USER='SCOTT';

no rows selected
```

7. Use the DBA\_PROFILES table to list the details of the ‘DEFAULT’ profile. Provide a screen shot below:

## CST8276 Lab 4: User Security

```
SQL> SELECT * FROM DBA_PROFILES WHERE PROFILE='DEFAULT';
```

PROFILE

RESOURCE\_NAME

RESOURCE

LIMIT

COM

---

DEFAULT

KERNEL

COMPOSITE\_LIMIT

UNLIMITED

NO

List 2 aspects of the default profile that could lead to performance issues:  
It initially defines unlimited resources

\_\_\_\_\_???

List 1 aspects of the default profile that could lead to security issues:

\_\_\_\_\_

8. In a sequence of “create user ....”, “alter user ...” and other privilege granting steps, you are to create a new user with the following criteria:
- Username - your last name (e.g., king)
  - Password – your last name (e.g., kingpwd)
  - Set the password to be expired.** (It will need to be changed on next connection.)
  - Use the DEFAULT profile.**
  - Use the default tablespaces (i.e., USERS and TEMP).
  - Use the “GRANT .... TO ....” command multiple times to grant CONNECT, RESOURCE and CREATE VIEW to the new user.

Show your work below:

```
SQL> CREATE USER ILGUN IDENTIFIED BY ILGUN;
```

User created.

```
SQL> ALTER USER ILGUN IDENTIFIED BY ILGUN PASSWORD EXPIRE;
```

User altered.

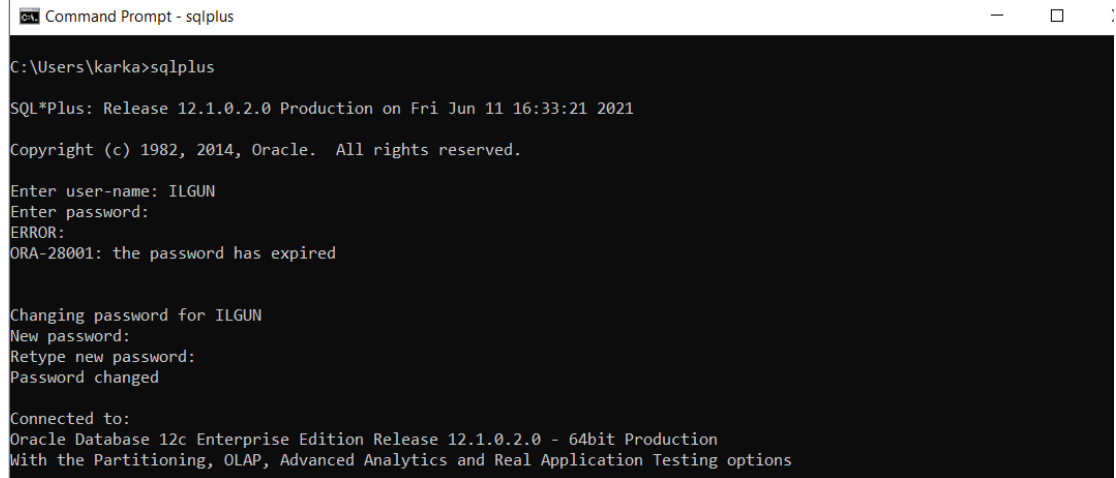
```
SQL> GRANT CONNECT, RESOURCE, CREATE VIEW TO ILGUN;
```

Grant succeeded.

```
SQL> _
```

## CST8276 Lab 4: User Security

- g. Keep your SQLPLUS session running as SYS AS DBA, and open a new command window (not as administrator). Then, logon to SQLPlus with your new account in the new window. You should get an 'expired password' message. When prompted enter *lastname*pwd as your “New” password. (It will let you reuse the original one....)



```
Command Prompt - sqlplus

C:\Users\karka>sqlplus

SQL*Plus: Release 12.1.0.2.0 Production on Fri Jun 11 16:33:21 2021

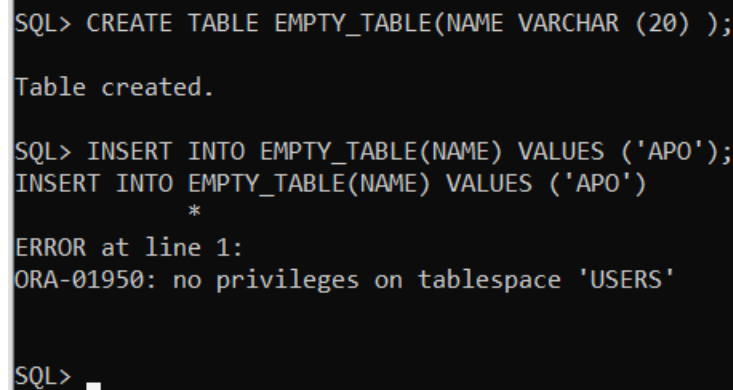
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Enter user-name: ILGUN
Enter password:
ERROR:
ORA-28001: the password has expired

Changing password for ILGUN
New password:
Retype new password:
Password changed

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

- h. Try to create a simple empty table from your new account. If successful, try to insert one record. What occurs? Show your work



```
SQL> CREATE TABLE EMPTY_TABLE(NAME VARCHAR (20));

Table created.

SQL> INSERT INTO EMPTY_TABLE(NAME) VALUES ('APO');
INSERT INTO EMPTY_TABLE(NAME) VALUES ('APO')
*
ERROR at line 1:
ORA-01950: no privileges on tablespace 'USERS'

SQL> _
```

- i. Keep your new account window open but change focus back to your original SYS as SYSBA window to alter the quota on the ‘USERS’ tablespace for your new user to be UNLIMITED.  
e.g., “ALTER USER myname QUOTA UNLIMITED ON USERS;” Show your work here:

## CST8276 Lab 4: User Security

```
SQL> ALTER USER ILGUN QUOTA UNLIMITED ON USERS;
User altered.
```

- j. Return to your new account in the other window and retry creating a new table and inserting a single row. Show your work here:

```
SQL> INSERT INTO NEW_TABLE(NAME) VALUES ('APO');
1 row created.
SQL>
```

**You're done.**