Using Oracle Database Vault with Oracle Database 12*c*

Activity Guide

D86597GC10

Edition 1.0

August 2014

Author

Maria Billings

Dominique Jeunot

Technical Contributors and Reviewers

Chi Ching Chui

Pat Huey

Yaping Li

Paul Needham

James Spiller

Sailaja Pasupuleti

Sravanti Tatiraju

Editor

Daniel Milne

Graphic Designer

Maheshwari Krishnamurthy

Publisher

Jayanthy Keshavamurthy

Table of Contents

Practices for Lesson 1: Introduction

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

In these practices, you will familiarize yourself with the computing environment used in this course and perform setup tasks:

- Enabling Unified Auditing
- Adding a Cloud Control Target

Note: Throughout these practices, **`Courier New bold`** is used to indicate command(s) that you enter. For example, the following indicates that you are to enter the **`date`** command:

```
$ date
Mon Jun 16 00:20:46 UTC 2014
$
```

Practice 1-1: Your Course Setup

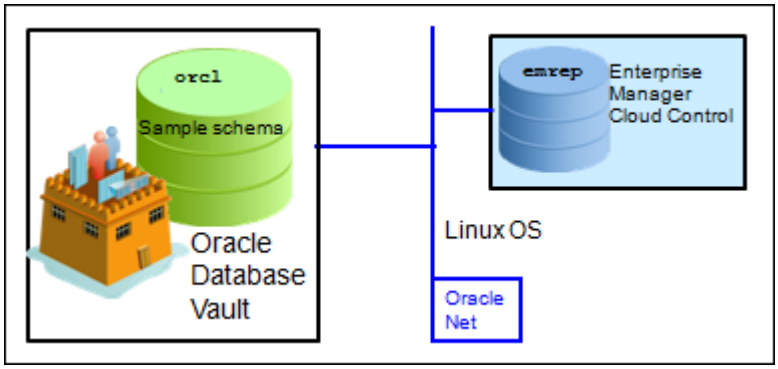
Overview

In this practice, you familiarize yourself with the computing environment used in this course. You make note of some important information that you will need when you perform the practices for this course. Fill in the Course Overview table as you gather the information.

Assumptions

You have a course setup on Linux OS as shown in this graphic with two database instances: `orcl` and `emrep`.

- Enterprise Manager Cloud Control is installed on `emrep`.
- An Oracle Database 12.1.0.1 with sample schema is on `orcl`.



Tasks

1. Log in to your assigned machine and open a terminal window: Right-click and select **Open in Terminal**.
2. Check your system date and time. Note it, especially if it is different from your own time zone.

```
$ date
Mon Jun 16 00:20:46 UTC 2014
$
```

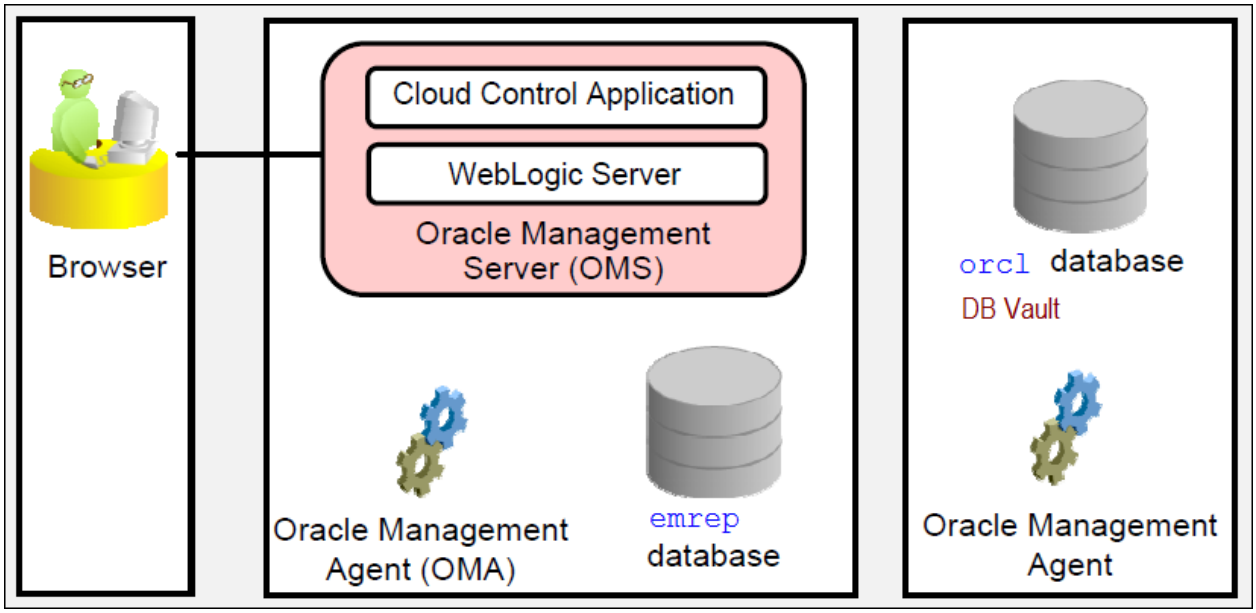
3. Review the running Oracle Process Monitor (`pmon`).

```
$ pgrep -lf pmon
29798 ora_pmon_orcl
30535 ora_pmon_emrep
$
```

   You should see that you have two instances running: `orcl` and `emrep`.

4. **Optional:** Explore the status of Cloud Control. Check whether the OMS and agent are up and running.

   Note: Perform this task only if you have plenty of time, because this step is entirely optional, while the next one is mandatory.




```
$ . oraenv
ORACLE_SID = [oracle] ? emrep
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ /u01/app/oracle/product/middleware/oms/bin/emctl status oms
Oracle Enterprise Manager Cloud Control 12c Release 4
Copyright (c) 1996, 2014 Oracle Corporation.  All rights
reserved.
WebTier is Up
Oracle Management Server is Up
$
$ /u01/app/oracle/product/agent12c/agent_inst/bin/emctl status
agent
Oracle Enterprise Manager Cloud Control 12c Release 4
Copyright (c) 1996, 2014 Oracle Corporation.  All rights
reserved.
------------------------------------------------------------
Agent Version          : 12.1.0.4.0
OMS Version            : 12.1.0.4.0
Protocol Version       : 12.1.0.1.0
Agent Home             :
/u01/app/oracle/product/agent12c/agent_inst
Agent Log Directory    :
/u01/app/oracle/product/agent12c/agent_inst/sysman/log
Agent Binaries         :
/u01/app/oracle/product/agent12c/core/12.1.0.4.0
Agent Process ID       : 2796
Parent Process ID      : 2747
```

```
Agent URL                : https://<your_hostname>:3872/emd/main/
Local Agent URL in NAT   : https://<your_hostname>:3872/emd/main/
Repository URL           :
https://<your_hostname>:4903/empbs/upload
Started at               : 2014-06-15 02:20:00
Started by user          : oracle
Operating System         : Linux version 2.6.39-
200.24.1.el6uek.x86_64 (amd64)
Last Reload              : (none)
Last successful upload                          : 2014-06-16
00:21:37
Last attempted upload                           : 2014-06-16
00:21:37
Total Megabytes of XML files uploaded so far : 2.47
Number of XML files pending upload           : 0
Size of XML files pending upload(MB)         : 0
Available disk space on upload filesystem    : 83.88%
Collection Status                               : Collections
enabled
```

```
Heartbeat Status                                : Ok
Last attempted heartbeat to OMS                 : 2014-06-16
00:21:16
Last successful heartbeat to OMS                : 2014-06-16
00:21:16
Next scheduled heartbeat to OMS                 : 2014-06-16
00:22:17

---------------------------------------------------------------
Agent is Running and Ready
$
```

5. **Required:** Set your environment variables for the ORCL database.

   Note: Each time you open a new terminal window, you must point to the database that you want to access. Using the `oraenv` utility is Oracle's recommended best practice.

```
$ . oraenv
ORACLE_SID = [emrep] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$
```

6. Discover and note your hostname. Your value will be different.

```
$ hostname --long
<your_hostname>
$
```

7. View your listener status and port number.

```
$ lsnrctl status
…
Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                14-JUN-2014 13:32:40
Uptime                    1 days 10 hr. 51 min. 37 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/<your_hostname>/listener/alert/log.
xml
Listening Endpoints Summary...
    (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<your_hostname>)(PORT=
1521)))
DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=<your_hostname>)(PORT=5
500))(Presentation=HTTP)(Session=RAW))
Services Summary...
Service "emrep" has 1 instance(s).
  Instance "emrep", status READY, has 1 handler(s) for this
service...
Service "emrepXDB" has 1 instance(s).
  Instance "emrep", status READY, has 1 handler(s) for this
service...
Service "orcl" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
Service "orclXDB" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
The command completed successfully
$
```

The port should be `1521`, but the other values may be different.

8. Find and note your IP address. Your values will be different.

   Tip: Enter `ping <your_hostname>` to start the ping test and press **Ctrl + C** to end it.

```
$ ping your_hostname
PING <your_hostname> (<your_IPAddress>) 56(84) bytes of data.
64 bytes from <your_hostname> (<your_IPAddress>): icmp_seq=1
ttl=64 time=0.014 ms
64 bytes from <your_hostname> (<your_IPAddress>): icmp_seq=2
ttl=64 time=0.015 ms
64 bytes from <your_hostname> (<your_IPAddress>): icmp_seq=3
ttl=64 time=0.019 ms
^C
--- <your_hostname> ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2967ms
rtt min/avg/max/mdev = 0.014/0.016/0.019/0.002 ms
$
```

Course Overview Table	
Item	**Value**
Your fully qualified hostname	*<your_hostname>*
Your IP address	*<your_IP_Address>*
Database name (SID) **Note:** This is also your service name.	orcl
Listener port	1521
Enterprise Manager (em) URL	https://localhost:7802/em
Username	**Password**
sysman	oracle_4U
sys	oracle_4U
system	oracle_4U
leo_dvowner	oracle_4U
bea_dvacctmgr	oracle_4U
dba_psmith	oracle_4U
hr	oracle_4U
oe	oracle_4U
bernst	oracle_4U
smavris	oracle_4U
kpartner	oracle_4U
wsmith	oracle_4U
ahunold	oracle_4U

Practice 1-2: Enabling Unified Auditing

Overview

In this practice, you enable Unified Auditing, if needed.

Assumptions

You completed the mandatory tasks of practice 1-1.

Tasks

1. **Required:** Log in to SQL*Plus as `sysdba` to check if Unified Auditing is enabled.

```
$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Mon Jun 16 00:27:55
2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.



Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> EXIT
$
```

   Unified Auditing is not enabled.

2. **Required:** Before you enable Unified Auditing, shut down all Oracle processes of all instances.
   a. Shut down the listener.

```
$ lsnrctl stop

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 16-JUN-
2014 00:33:01

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
The command completed successfully
$
```

b. Shut down the `orcl` instance.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

c. Shut down the `emrep` instance.

a) Stop the OMS.

```
$ cd /u01/app/oracle/product/middleware/oms
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms
$ $OMS_HOME/bin/emctl stop oms
Oracle Enterprise Manager Cloud Control 12c Release 4
Copyright (c) 1996, 2014 Oracle Corporation.  All rights
reserved.
Stopping WebTier...
WebTier Successfully Stopped
Stopping Oracle Management Server...
Oracle Management Server Successfully Stopped
Oracle Management Server is Down
$
```

b) Shut down the `emrep` repository database instance.

```
$ . oraenv
ORACLE_SID = [orcl] ? emrep
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> shutdown immediate
Database closed.
```

```
Database dismounted.
ORACLE instance shut down.
SQL> EXIT
$
```

d. Verify that all instances are down.

```
$ pgrep -lf pmon
$
```

3. **Required:** Enable Unified Auditing. Be cautious to copy the whole `make` command with the **`ORACLE_HOME=$ORACLE_HOME`** argument.

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk uniaud_on ioracle
ORACLE_HOME=$ORACLE_HOME
/usr/bin/ar d
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a
kzanang.o
/usr/bin/ar cr
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/kzaiang.o

chmod 755 /u01/app/oracle/product/12.1.0/dbhome_1/bin

 - Linking Oracle
rm -f /u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/orald  -o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle -m64 -z
noexecstack -Wl,--disable-new-dtags -
L/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ -
L/u01/app/oracle/product/12.1.0/dbhome_1/lib/ -
L/u01/app/oracle/product/12.1.0/dbhome_1/lib/stubs/   -Wl,-E
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/opimai.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ssoraed.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/ttcsoi.o -Wl,-
-whole-archive -lperfsrv12 -Wl,--no-whole-archive
/u01/app/oracle/product/12.1.0/dbhome_1/lib/nautab.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naeet.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naect.o
/u01/app/oracle/product/12.1.0/dbhome_1/lib/naedhs.o
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/config.o  -
lserver12 -lodm12 -lcell12 -lnnet12 -lskgxp12 -lsnls12 -lnls12
-lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -lnls12 -lxml12 -
lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -lnls12 -lclient12  -
lvsn12 -lcommon12 -lgeneric12 -lknlopt `if /usr/bin/ar tv
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a |
grep xsyeolap.o > /dev/null 2>&1 ; then echo "-loraolap12" ; fi`
-lskjcx12 -lslax12 -lpls12  -lrt -lplp12 -lserver12 -lclient12
-lvsn12 -lcommon12 -lgeneric12 `if [ -f
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libavserver12.a ] ;
then echo "-lavserver12" ; else echo "-lavstub12"; fi` `if [ -f
```

```
/u01/app/oracle/product/12.1.0/dbhome_1/lib/libavclient12.a ] ;
then echo "-lavclient12" ; fi` -lknlopt -lslax12 -lpls12  -lrt -
lplp12 -ljavavm12 -lserver12  -lwwg  `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnro12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnnzst12 -lzt12 -
lztkg12 -lmm -lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -
lcore12 -lsnls12 -lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -
lnls12 -lcore12 -lnls12 -lztkg12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnro12 `cat
/u01/app/oracle/product/12.1.0/dbhome_1/lib/ldflags`    -
lncrypt12 -lnsgr12 -lnzjs12 -ln12 -lnl12 -lnnzst12 -lzt12 -
lztkg12   -lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -
lsnls12 -lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -
lcore12 -lnls12 `if /usr/bin/ar tv
/u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/libknlopt.a |
grep "kxmnsd.o" > /dev/null 2>&1 ; then echo " " ; else echo "-
lordsdo12"; fi` -
L/u01/app/oracle/product/12.1.0/dbhome_1/ctx/lib/ -lctxc12 -
lctx12 -lzx12 -lgx12 -lctx12 -lzx12 -lgx12 -lordimt12 -lclsra12
-ldbcfg12 -lhasgen12 -lskgxn2 -lnnzst12 -lzt12 -lxml12 -locr12 -
locrb12 -locrutl12 -lhasgen12 -lskgxn2 -lnnzst12 -lzt12 -lxml12
-lgeneric12 -loraz -llzopro -lorabz2 -lipp_z -lipp_bz2 -
lippdcemerged -lippsemerged -lippdcmerged  -lippsmerged -
lippcore  -lippcpemerged -lippcpmerged  -lsnls12 -lnls12  -
lcore12 -lsnls12 -lnls12 -lcore12 -lsnls12 -lnls12 -lxml12 -
lcore12 -lunls12 -lsnls12 -lnls12 -lcore12 -lnls12 -lsnls12 -
lunls12  -lsnls12 -lnls12  -lcore12 -lsnls12 -lnls12 -lcore12 -
lsnls12 -lnls12 -lxml12 -lcore12 -lunls12 -lsnls12 -lnls12 -
lcore12 -lnls12 -lasmclnt12 -lcommon12 -lcore12  -laio -lons
`cat /u01/app/oracle/product/12.1.0/dbhome_1/lib/sysliblist` -
Wl,-rpath,/u01/app/oracle/product/12.1.0/dbhome_1/lib -lm
`cat /u01/app/oracle/product/12.1.0/dbhome_1/lib/sysliblist` -
ldl -lm   -L/u01/app/oracle/product/12.1.0/dbhome_1/lib
test ! -f /u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle ||\
       mv -f /u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracleO
mv /u01/app/oracle/product/12.1.0/dbhome_1/rdbms/lib/oracle
/u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
chmod 6751 /u01/app/oracle/product/12.1.0/dbhome_1/bin/oracle
$
```

4. **Required:** Restart the processes and verify that Unified Auditing is now enabled.
   a. Restart the database instances, both `EMREP` and `ORCL`.

```
$ . oraenv
ORACLE_SID = [emrep] ? emrep
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
```

```
$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Mon Jun 16 00:39:47
2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.
SQL> STARTUP
ORACLE instance started.

Total System Global Area 1286066176 bytes
Fixed Size                  2287960 bytes
Variable Size             452986536 bytes
Database Buffers          822083584 bytes
Redo Buffers                8708096 bytes
Database mounted.
Database opened.
SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
$
```

```
$ . oraenv
ORACLE_SID = [emrep] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.1.0 Production on Mon Jun 16 00:42:03
2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

Connected to an idle instance.
SQL> STARTUP
ORACLE instance started.

ORACLE instance started.

Total System Global Area  501059584 bytes
```

```
Fixed Size                  2290024 bytes
Variable Size             268439192 bytes
Database Buffers          222298112 bytes
Redo Buffers                8032256 bytes
Database mounted.
Database opened.
SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
$
```

b. Restart the listener.

```
$ lsnrctl start

LSNRCTL for Linux: Version 12.1.0.1.0 - Production on 16-JUN-
2014 00:44:29

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr:
please wait...

TNSLSNR for Linux: Version 12.1.0.1.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Log messages written to
/u01/app/oracle/diag/tnslsnr/EDRSR44P1/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edRSr44p1.us.oracle.co
m)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.1.0
- Production
Start Date                16-JUN-2014 00:44:29
Uptime                    0 days 0 hr. 0 min. 0 sec
```

```
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.o
ra
Listener Log File
/u01/app/oracle/diag/tnslsnr/EDRSR44P1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edRSr44p1.us.oracle.co
m)(PORT=1521)))
The listener supports no services
The command completed successfully
$
```

c. Restart OMS.

```
$ export OMS_HOME=/u01/app/oracle/product/middleware/oms
$ $OMS_HOME/bin/emctl start oms
Oracle Enterprise Manager Cloud Control 12c Release 4
Copyright (c) 1996, 2014 Oracle Corporation.  All rights
reserved.
Starting Oracle Management Server...
Starting WebTier...
WebTier Successfully Started
Oracle Management Server Successfully Started
Oracle Management Server is Up
$
```

d. Verify that Unified Auditing is enabled. You can see, in the SQL*Plus banner and in the V$OPTION view, that Unified Auditing is enabled.

```
$ sqlplus / as sysdba

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, Oracle Label Security, OLAP, Advanced
Analytics, Real Application Testing and Unified Auditing options

SQL> COL  parameter FORMAT A20
SQL> COL  value     FORMAT A20
SQL> select parameter , value
     from v$option
     where PARAMETER = 'Unified Auditing';
```

```
  2    3
PARAMETER            VALUE
-------------------- --------------------
Unified Auditing     TRUE

SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics, Real
Application Testing and Unified Auditing options
$
```

Note: The SQL banner and space lines are removed in this activity guide, to avoid output that is not the main focus of a practice.

Practice 1-3: Adding a Cloud Control Target

Overview

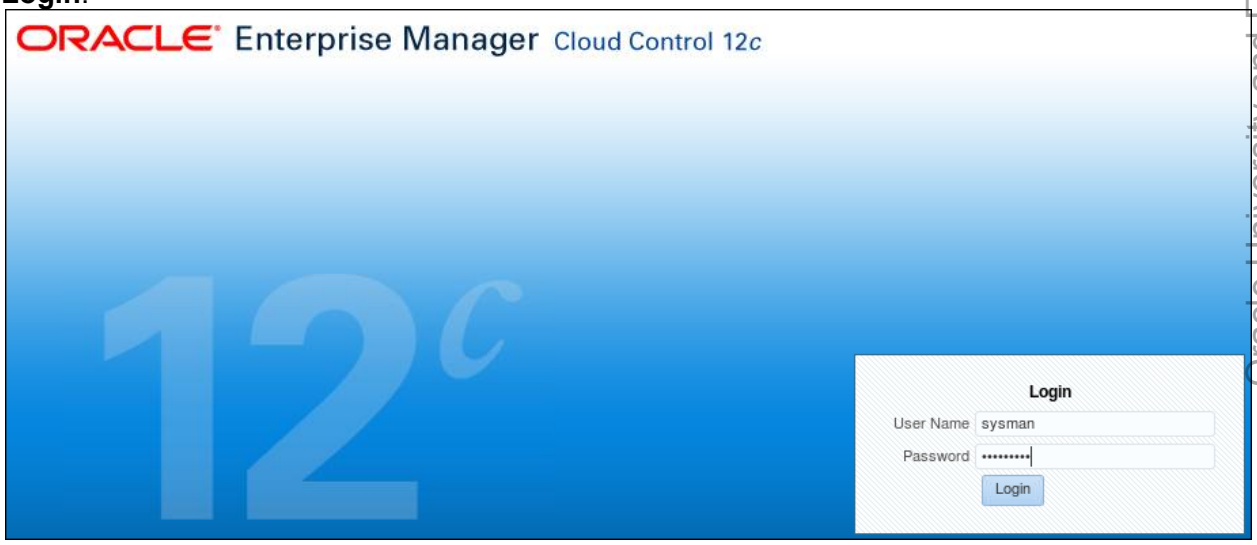
In this practice, you add the `orcl` database instance as a new target to be monitored by Oracle Enterprise Manager Cloud Control.

Assumptions

The databases and Cloud Control are up-and-running.

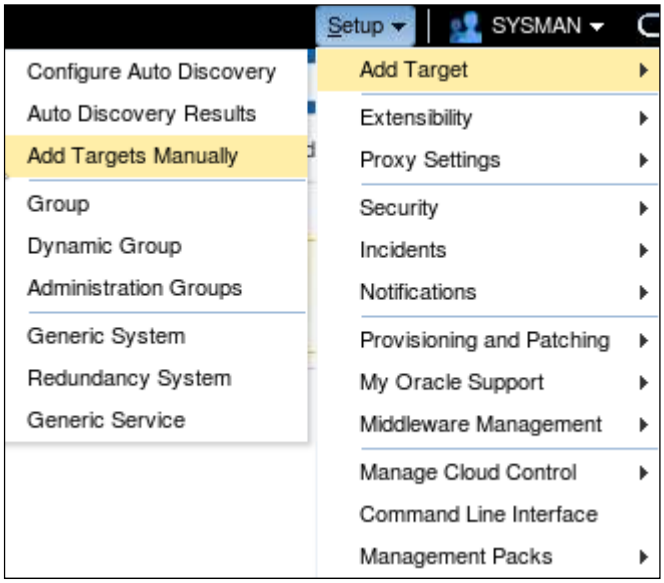
Tasks

1. **Required:** Add the `orcl` database instance as a new target to be monitored by Oracle Enterprise Manager Cloud Control.
   a. Start a web browser with **`https://<your hostname>:7802/em`** or simply **`https://localhost:7802/em`**
   b. The first time you start the **https://localhost:7802/em**, you may receive a Secure Connection Failed message and you will need to add a security exception. If so, click **Or you can add an exception**.
   c. At the end of the alert box, click **I Understand the Risks**.
   d. At the bottom of the page, click **Add Exception**.
   e. In the Add Security Exception pop-up window, click **Get Certificate**.
   f. Confirm that **Permanently store this exception** is selected in your training environment, and click **Confirm Security Exception**.
   g. Log in to Enterprise Manager Cloud Control as the `sysman` user with the `oracle_4U` password.
   h. Enter `sysman` as the username and `oracle_4U` as the password, and then click **Login**.

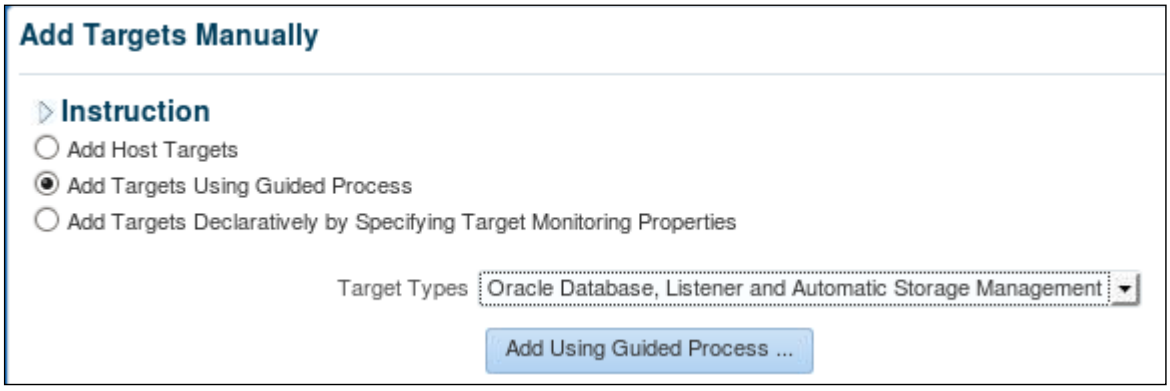


Note: If the License Agreement page appears, click **I Accept**.

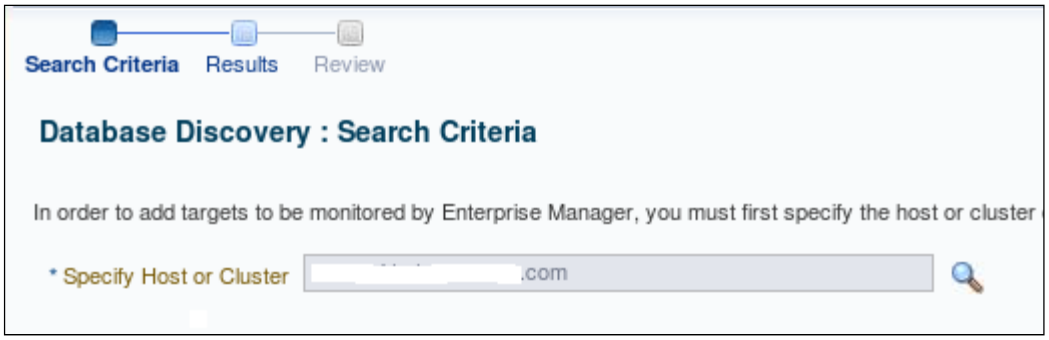
   i. From the Setup menu (at the top right), select **Add Target > Add Targets Manually**.



j. On the Add Targets Manually page, select **Add Targets Using Guided Process**. Then, from the Target Types drop-down list, select **Oracle Database, Listener and Automatic Storage Management**. Then click the **Add Using Guided Process** button.



k. On the Database Discovery: Search Criteria page, next to the "Specify Host or Cluster" field, click the magnifying glass to find your host. Select your host, and then click **Select**. Back on the Search Criteria page, click **Next**.

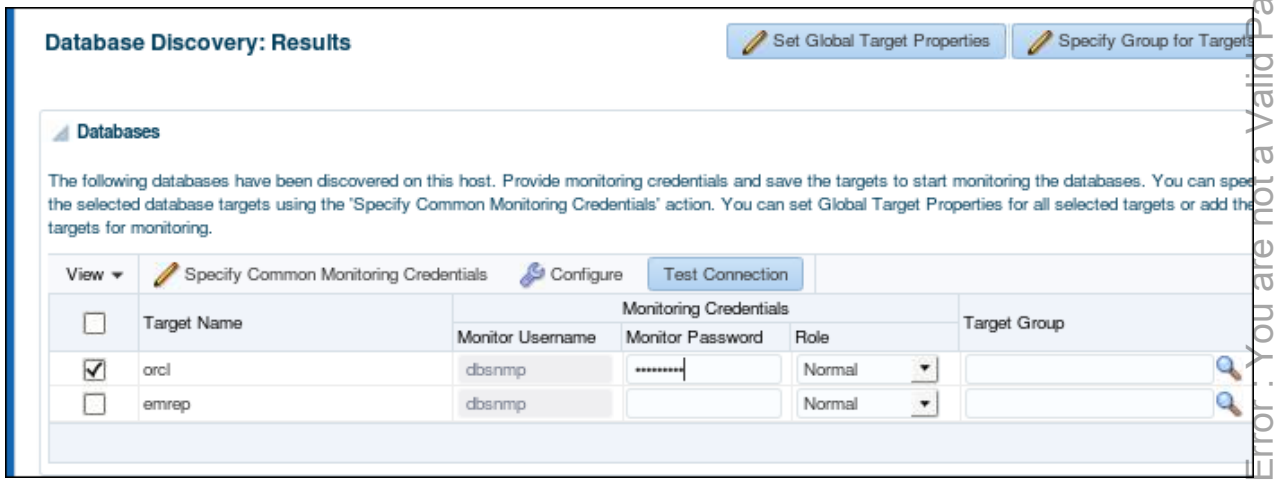


1) On the Database Discovery: Results page, in the Databases list, select the **orcl** database.
2) Unlock the DBSNMP user. This user is the monitoring user used to test the connection once the target is being added. Open a terminal window.

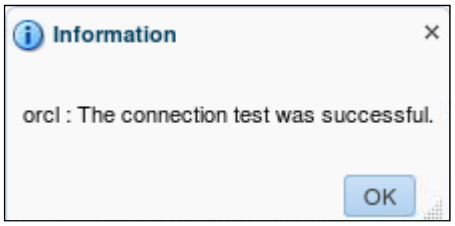
```
$ . oraenv
```

```
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL> alter user dbsnmp identified by oracle_4U account unlock;

User altered.

SQL> EXIT
$
```

3) Continue in Cloud Control. Enter `oracle_4U` for in the Monitor Password field.



4) Click the **Test Connection** button. You should receive the following message:



5) Click the **OK** button, the **Next** button, and then the **Save** button to complete the operation, and finally click the **Close** button.

Practices for Lesson 2: Database Vault Overview

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

In these practices, you will test your understanding of the topics presented in the lesson, and then you will view a video to gain a high-level overview of Database Vault and its main concepts.

Practice 2-1: Testing Your Knowledge

Overview

In this practice, you answer questions that are designed to test your understanding of the topics covered in this lesson. Your instructor will provide possible answers.

Tasks

1. Choose the correct definition for the following from the options provided below the list of components.
   - ____ Realm
   - ____ Rule set
   - ____ Command rule
   - ____ Secure application role
   - ____ Factor

   a. A role that can be activated by a session if specific conditions are true
   b. A collection of rules that are evaluated to determine access
   c. An attribute of a database session that can be referenced by Database Vault components to help determine access
   d. A boundary defined to protect database object
   e. Specific conditions that must be met for a given SQL command to be executed

2. Identify whether the following statements are true or false:
   a. Database Vault provides additional security without requiring changes to applications.
   b. Database Vault can be used to restrict DBAs from seeing the data in database tables.
   c. The Database Vault components administration is accessed using Oracle Enterprise Manager Cloud Control.

Practice 2-2: Viewing the "Quick Start Tutorial: Securing a Schema from DBA Access" Video

Overview

In this practice, you watch the "Quick Start Tutorial: Securing a Schema from DBA Access" video, which provides a visual overview of Database Vault functionality.

Assumptions

The videos are included in the course setup. Access to the HTML version is described below.

Note: You can also view these videos after the course via the Oracle Learning Chanel (in MP4 format).

Tasks

1. Double-click the **Oracle's Home** icon on your desktop.
2. Double-click the **demos** folder and then the **dv_quick_start** folder.
3. In an Oracle classroom, double-click the HTML version (`dv_quick_start.html`) to start the video. In other environments, you may use the MP4 version of the same video.

   Note: Audio is available via the Oracle Learning Chanel, but not in most classrooms.
4. Use the controls in the center and at the bottom of the presentation window to start, pause, and stop the video, as suits your personal learning style.
5. Uninterrupted viewing of this demonstration takes about four minutes. When you have finished viewing the presentation, close your web-browser window.

Practices for Lesson 3: Configuring Database Vault

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

In these practices, you will enable and configure Database Vault and some test users. You must complete the mandatory part of this practice, because all following practices depend on the Database Vault configuration.

Practice 3-1: Configuring Database Vault

Overview

In this **mandatory** practice, you configure Database Vault in Oracle Database 12c for demo purposes. You implement separation of duties by creating separate user accounts: `leo_dvowner` for the `DV_OWNER` role and `bea_dvacctmgr` for the `DV_ACCTMGR` role.

Assumptions

Database Vault is installed in the Oracle Database 12c database.

Tasks

1. Log in to SQL*Plus as SYSDBA. (If you would like to confirm how to do this or view the entire output, revisit Practice 1-1 steps 5 and 9.)

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba
SQL>
```

2. Set up Oracle user accounts that are used throughout this course:
   - Create the `leo_dvowner` user with the `CREATE SESSION` and `SELECT ANY DICTIONARY` privileges and the `DV_OWNER` role.
   - Create the `bea_dvacctmgr` user with the `CREATE SESSION` privilege and the `DV_ACCTMGR` role.

```
SQL> create user leo_dvowner   identified by oracle_4U;

User created.

SQL> create user bea_dvacctmgr identified by oracle_4U;

User created.

SQL> grant create session to leo_dvowner , bea_dvacctmgr;

Grant succeeded.

SQL> grant select any dictionary to leo_dvowner;

Grant succeeded.

SQL>
```

3. Configure Database Vault, specifying the top Oracle user accounts. This is a one-time task in the lifetime of a database.

```
SQL> exec DVSYS.CONFIGURE_DV (-
                  dvowner_uname => 'leo_dvowner', -
                  dvacctmgr_uname => 'bea_dvacctmgr')
> >
PL/SQL procedure successfully completed.

SQL>
```

Note: Oracle recommends **not** giving these accounts to individuals for regular Database Vault administration tasks, but to keep them as "backup" for the highest-level tasks. You should grant the `DV_OWNER` role to a trusted employee and the `DV_ACCTMGR` role, ideally, to a different trusted employee.

4. Log in to SQL*Plus as a user with the `DV_OWNER` role. This course uses `leo_dvowner`.

```
SQL> connect leo_dvowner
Password:
Connected.
SQL>
```

5. Enable Database Vault.

```
SQL> exec DVSYS.DBMS_MACADM.ENABLE_DV

PL/SQL procedure successfully completed.

SQL>
```

Note: Users with the `DV_OWNER` role can enable and disable Database Vault according to the organizational needs.

6. Configuration changes should be followed by a database restart. Use `SYSDBA` to shut down and start up the `ORCL` database and confirm that the `Oracle Database Vault` parameter is set to `TRUE`.

```
SQL> connect / as sysdba
Connected.
SQL> COL parameter FORMAT A30
SQL> COL value FORMAT A10
SQL> select * from v$option where parameter = 'Oracle Database
Vault';

PARAMETER                      VALUE      CON_ID
------------------------------ ---------- ----------
Oracle Database Vault          FALSE               0

SQL>
```

Note: Before the database restart, the parameter was `FALSE`.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area  501059584 bytes
Fixed Size                  2290024 bytes
Variable Size             268439192 bytes
Database Buffers          222298112 bytes
Redo Buffers                8032256 bytes
Database mounted.
Database opened.
SQL> select * from v$option
     where parameter = 'Oracle Database Vault';
  2
PARAMETER                      VALUE          CON_ID
------------------------------ ---------- ----------
Oracle Database Vault               TRUE               0

SQL> exit
$
```

Practice 3-2: Setting Up Practice Accounts

Overview

In this **mandatory** practice, you create the roles and database users that are needed throughout these practices. They represent typical user definitions as might exist in an office that has an `HR` application (based on the `HR` schema) and a sales application (based on the `OE` schema).

Assumptions

The `HR` schema is unlocked.

Tasks

1. Execute the **`create_users.sql`** script (located in the `labs` directory) to create the users and roles that you will use in these practices.

   The following are the descriptions of the users and roles that are created. You may want to refer to this list throughout the course.

   - **`SMAVRIS`:** `HR` Manager
   - **`WSMITH`:** `HR` Clerk
   - **`KPARTNER`:** `HR` Clerk
   - **`BERNST`:** `HR` Application DBA (has the `HR_DBA` role)
   - **`AHUNOLD`:** Order Entry Application DBA (has the `OE_DBA` role)
   - **`DBA_PSMITH`**: `Oracle System Privilege and Role Management Realm` authorization and `DBA` role

```
$ cd ~/labs
$ sqlplus /nolog @lab_03_02_01.sql
$
```

Practice 3-3: Viewing Configuration Videos

Overview

In this optional practice, the videos provide a visual overview of two different types of configuration:

- "Configuring Database Vault in Oracle Database 12c" presents the one-time Database Vault configuration that you executed in Practice 3-1.
- "Configuring Database Vault Users in Cloud Control 12c" shows a task that you need to perform for all existing Oracle users that need access to Database Vault, by using the graphical interface of Enterprise Manager Cloud Control.

Assumptions

The videos are included in the course setup. Access to the HTML version is described below.

Note: You can also these videos after the course via the Oracle Learning Chanel (in MP4 format).

Tasks

1. To view the "Configuring Database Vault in Oracle Database 12c" video, double-click the **Oracle's Home** icon on your desktop.
2. Double-click the **demos** folder and then the **dv01_configure** folder.
3. In an Oracle classroom, double-click the HTML version (`dv01_configure.html`) to start the video. In other environments, you may use the MP4 version of the same video.

   Note: Audio is available via the Oracle Learning Chanel, but not in most classrooms.
4. Use the controls in the center and at the bottom of the presentation window to start, pause, and stop the video, as suits your personal learning style.
5. Uninterrupted viewing of this demonstration takes about three minutes.
6. To view the *Configuring Database Vault Users in Cloud Control 12c* video, from the **demos** folder, click the **dv_cc_user** folder.
7. In an Oracle classroom, double-click the HTML version (`dv_cc_user.html`) to start the video. In other environments, you may use the MP4 version of the same video.

   Note: Audio is available via the Oracle Learning Chanel, but not in most classrooms.
8. Use the controls in the center and at the bottom of the presentation window to start, pause, and stop the video, as suits your personal learning style.
9. Uninterrupted viewing of this demonstration takes about three minutes. When you have finished viewing the presentation, close your web-browser window.

Practice 3-4: Configuring a Database Vault User in Cloud Control 12*c*

Overview

In this **mandatory** practice, you configure the `leo_dvowner` user to have access to Database Vault via Cloud Control. The task steps are similar to what was demonstrated in the "Configuring Database Vault Users in Cloud Control 12c" video.

Assumptions

The `leo_dvowner` user was created in Practice 3-1.

Tasks

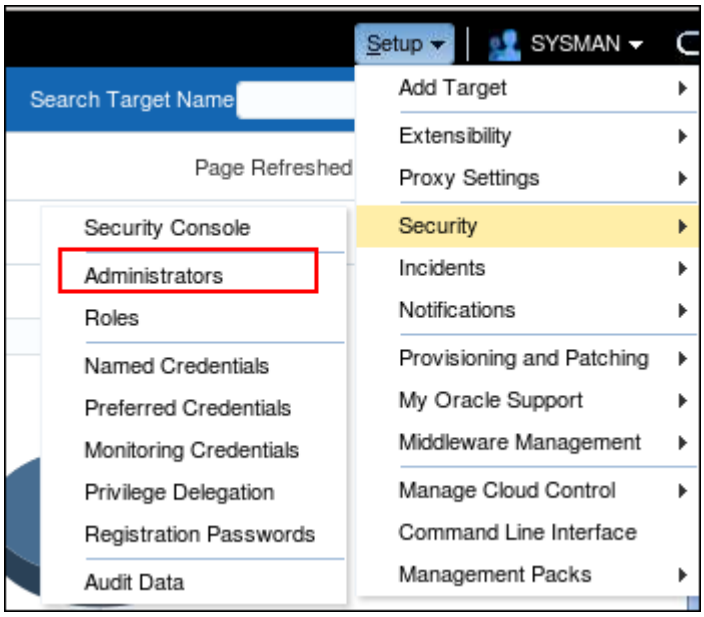
1. If you logged out of EM Cloud Control at the end of Practice 1-1, log in to Enterprise Manager Cloud Control as the `sysman` user with the `oracle_4U` password.
2. Set up the `leo_dvowner` user as Cloud Control administrator with the `EM_USER` and `PUBLIC` roles, and add the `ORCL` target with the `CONNECT` privilege.

   If you want to see more details than the following high-level steps provide, review the *Configuring Database Vault Users in Cloud Control 12c* video.

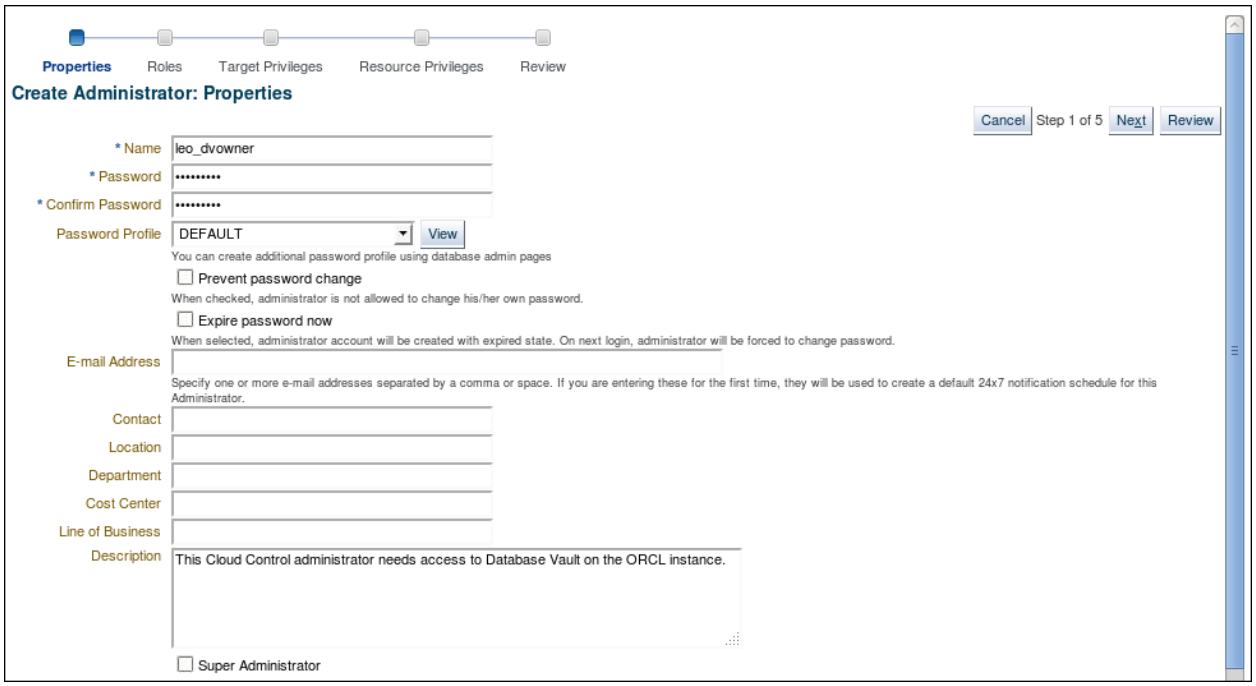
   a. From the Setup menu (at the top right), select **Security > Administrators**.
   b. Click the **Create** button and enter the information for the `leo_dvowner` who is NOT a "Superuser." Set `oracle_4U` as the password. Click **Next**.
   c. Accept the `EM_USER` and `PUBLIC` roles. Click **Next**.
   d. On "Create Administrator leo_dvowner: Target Privileges" page, scroll down and click **Add** in the Target Privileges section. Select the **ORCL** database instance target from the drop-down list and click **Select**.
   e. In the Manage Target Privilege Grants column, click the pen next to View.
   f. Select **Connect Target** and click **Continue**.
   g. Scroll to the bottom to confirm that the `Connect Target` privilege is specified for the `orcl` database instance. Then click **Review**.
   h. On the Create Administrator leo_dvowner: Review page, confirm your specifications and then click **Finish** to complete the creation of the `leo_dvowner` user as Cloud Control administrator.

Tip: Compare your screen with the following screenshots to ensure that this task is correctly completed.

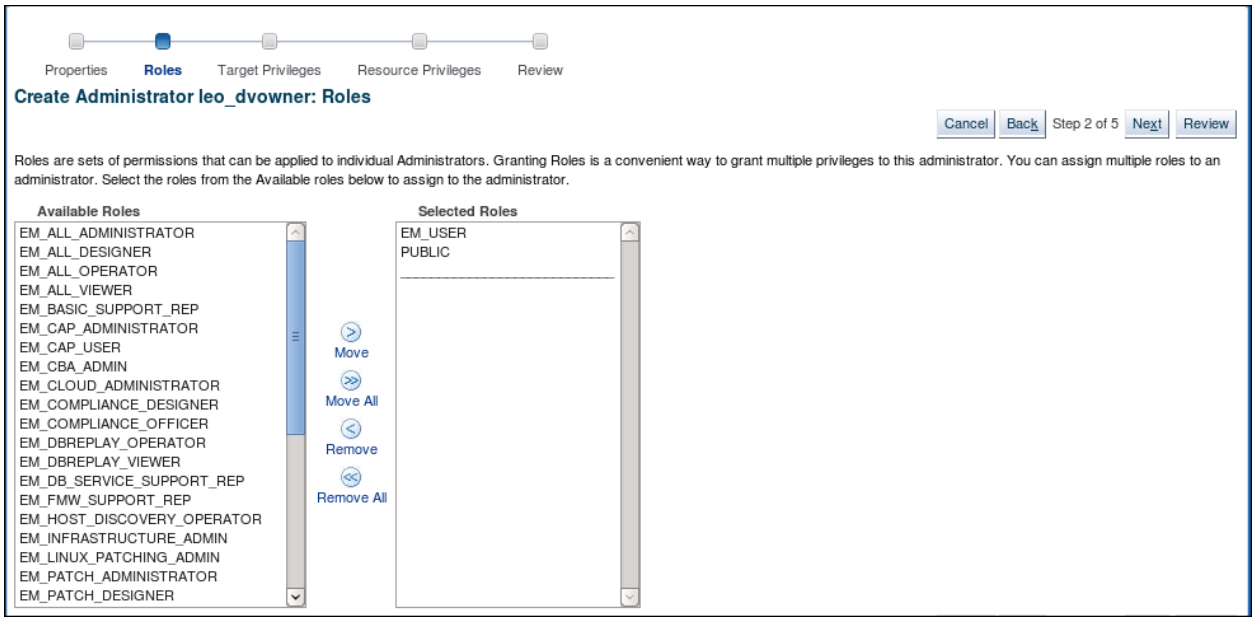
a.



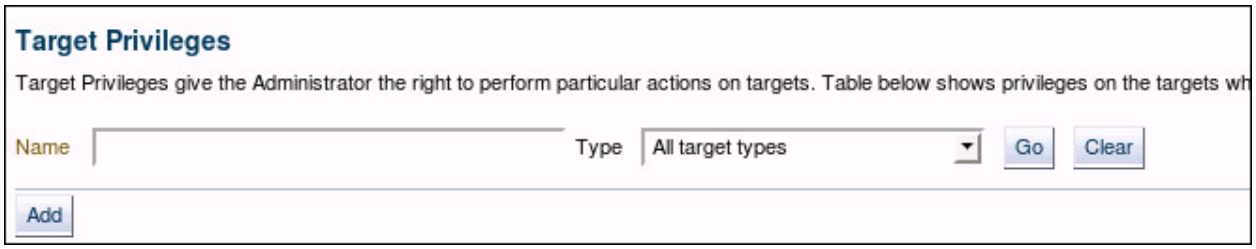
b.



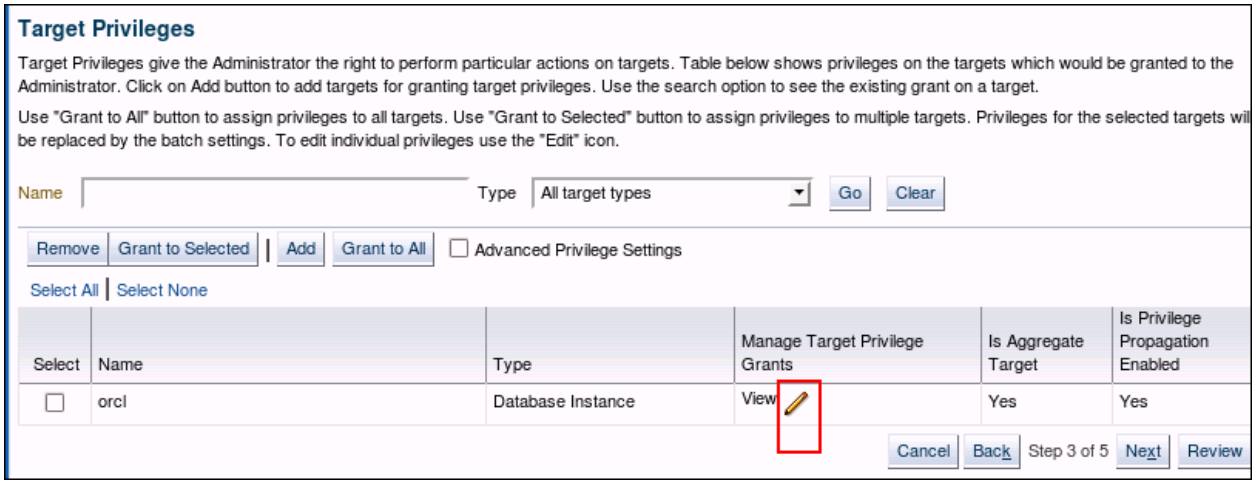
c.



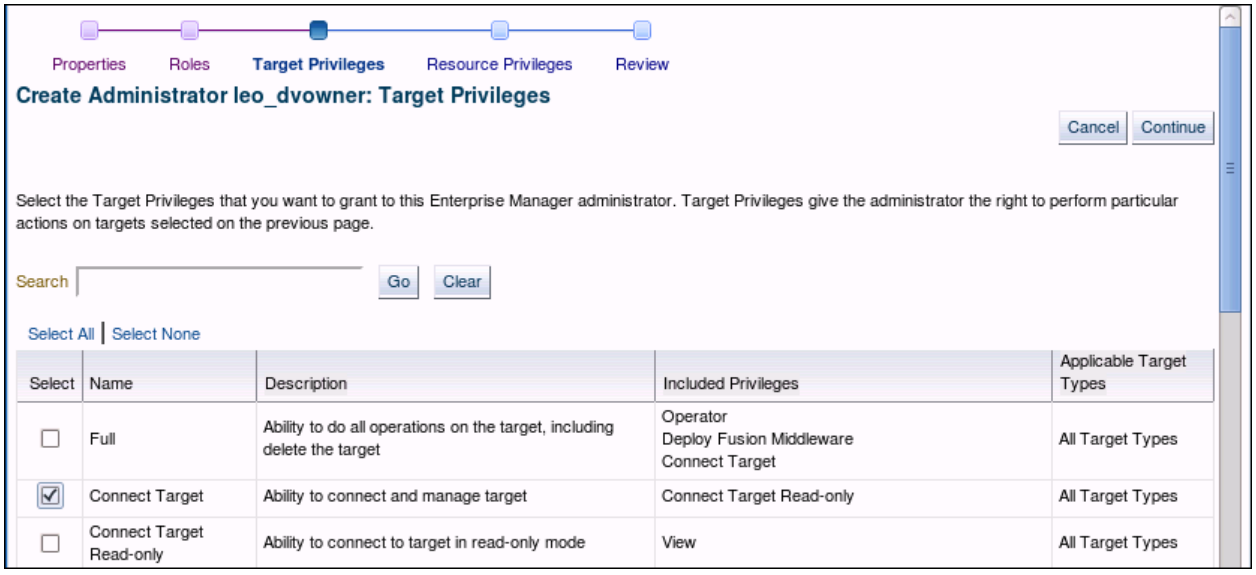
d.



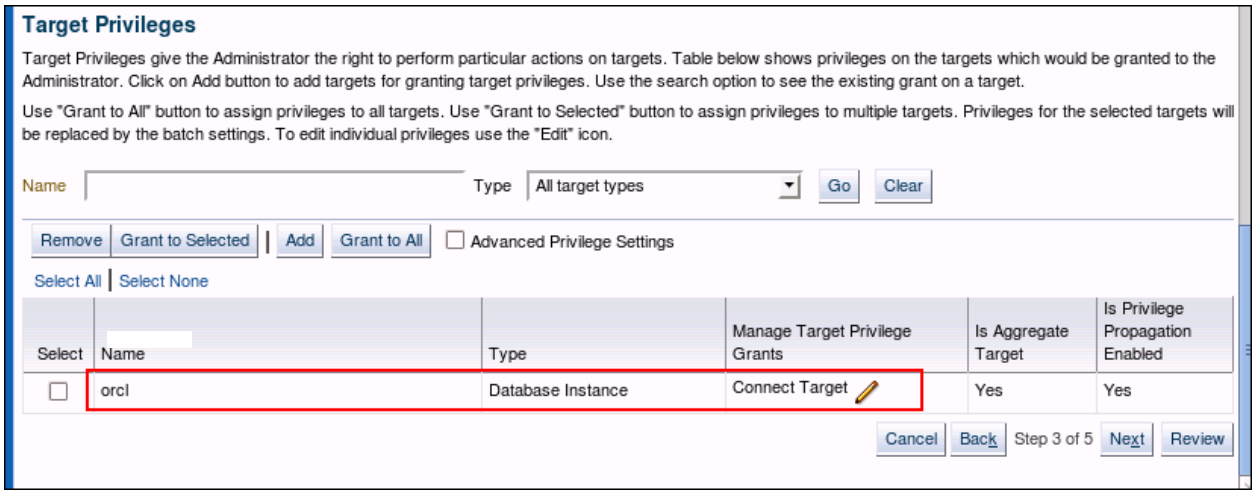
e.



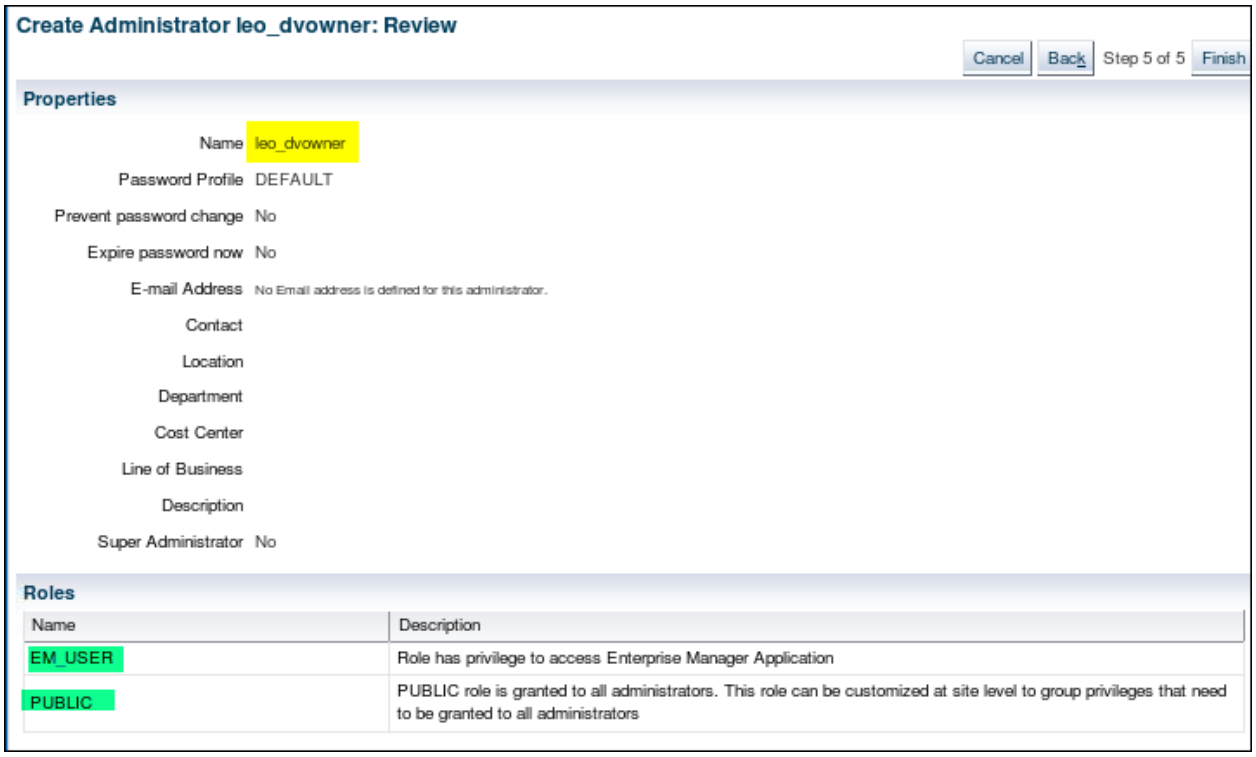
f.

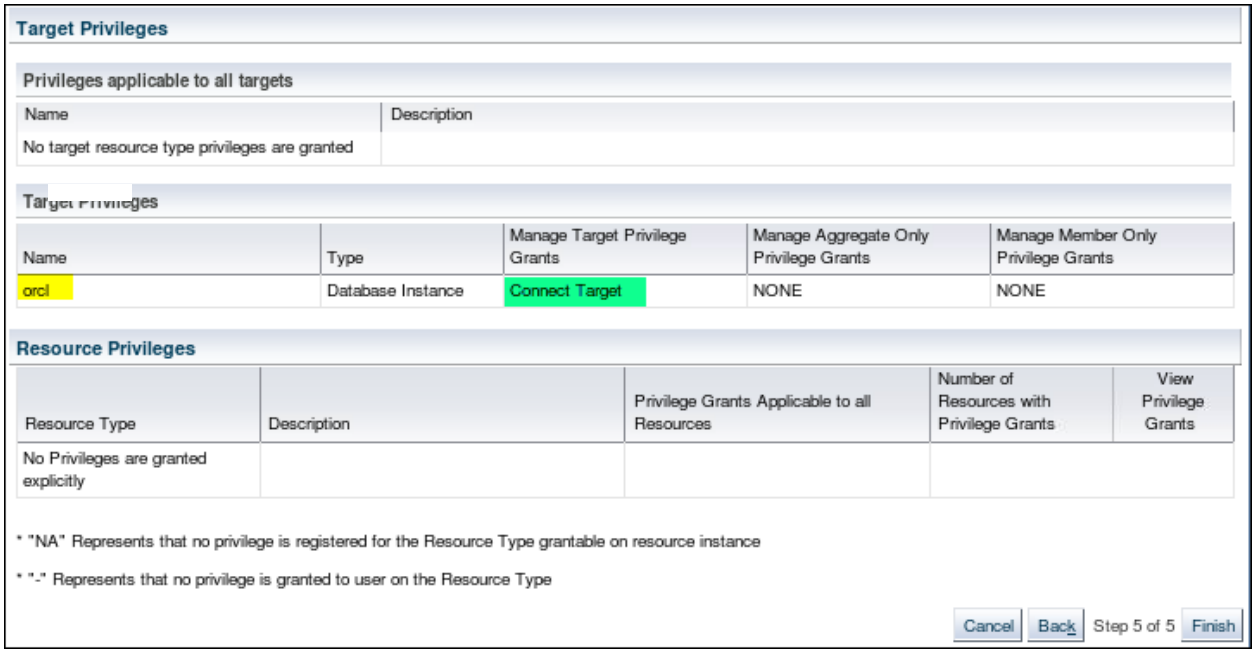


g.

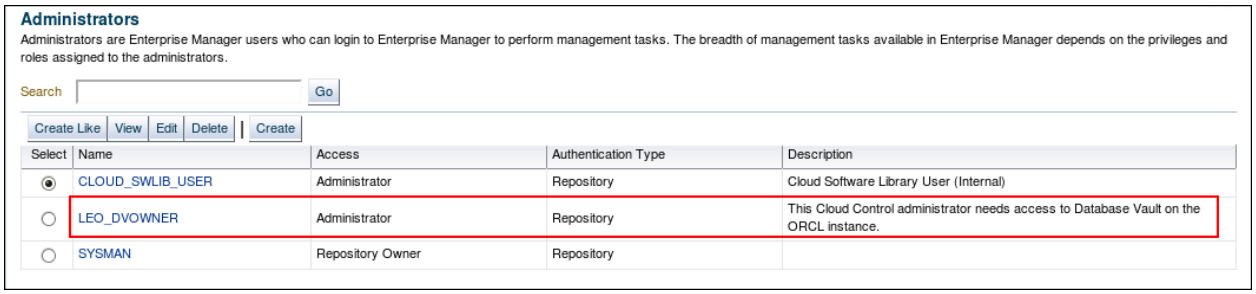


h.





Success:

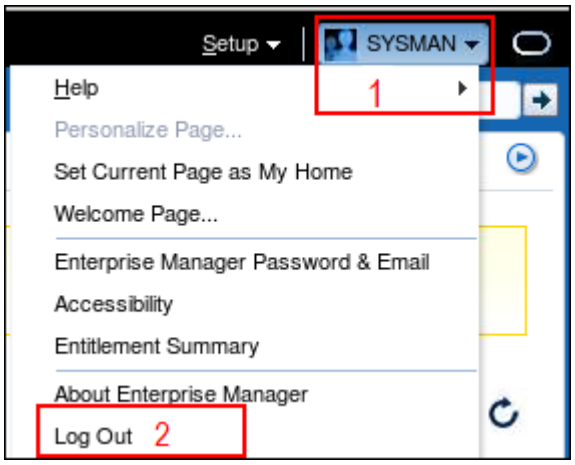


3. To test access, log out of Enterprise Manager Cloud Control and all targets and log in as the `leo_dvowner` user.
   a. From the `SYSMAN` menu (at the top right), select **Logout**.
   b. Enter `leo_dvowner` as **User Name** and `oracle_4U` as **Password**, and click **Login**.
   c. This is the very first time that the `leo_dvowner` user logs in to Enterprise Manager Cloud Control. The Accessibility Preference page appears. The “Your accessibility preferences are presented because this is your first login. You can set these now, or at anytime by using Username menu.” message appears. Click **I’ll deal with this later**.
   d. If you are offered the "Welcome to Enterprise Manager Cloud Control 12c" page, select **Databases** as the home page for `leo_dvowner`.
   e. Navigate to **Targets > Databases** use either the **Search List** to select the orcl target database.
   f. From the Security menu, select **Database Vault**.
   g. Enter `leo_dvowner` as **Username** and `oracle_4U` as **Password** to log in to the ORCL database.

   Note: If you save the credentials you do not need to enter them each time.

Tip: Compare your screen with the following screenshots to ensure that this task is correctly completed.

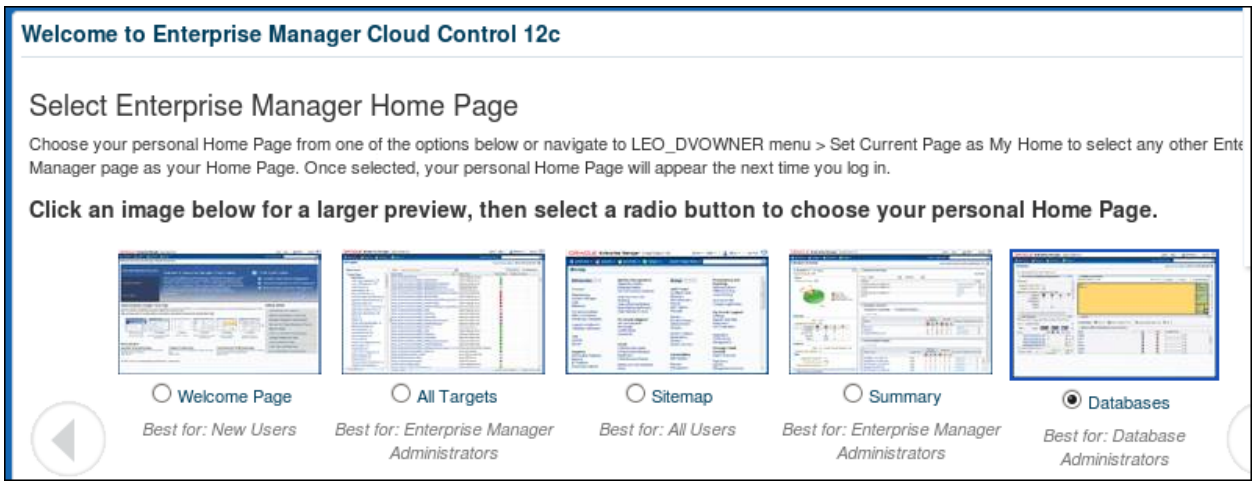
a.



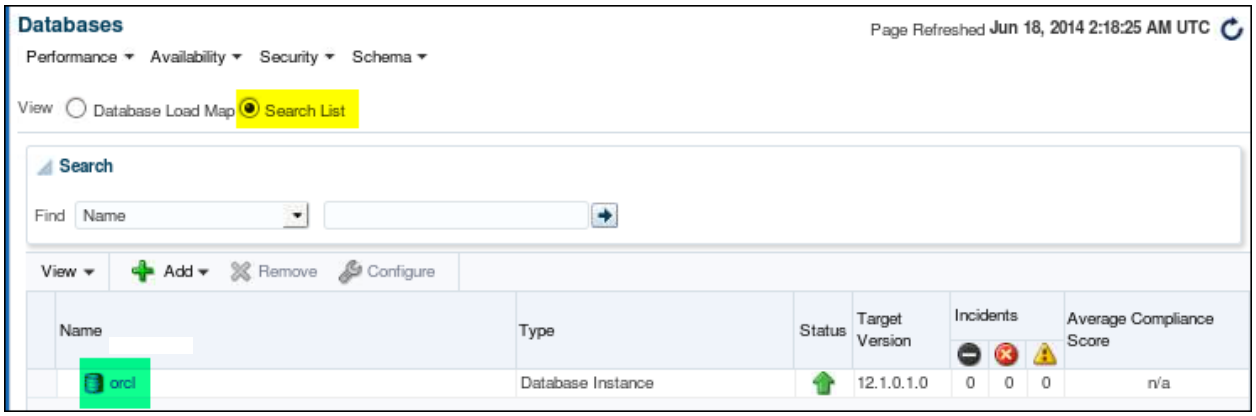
b.

c. Step c is not show.

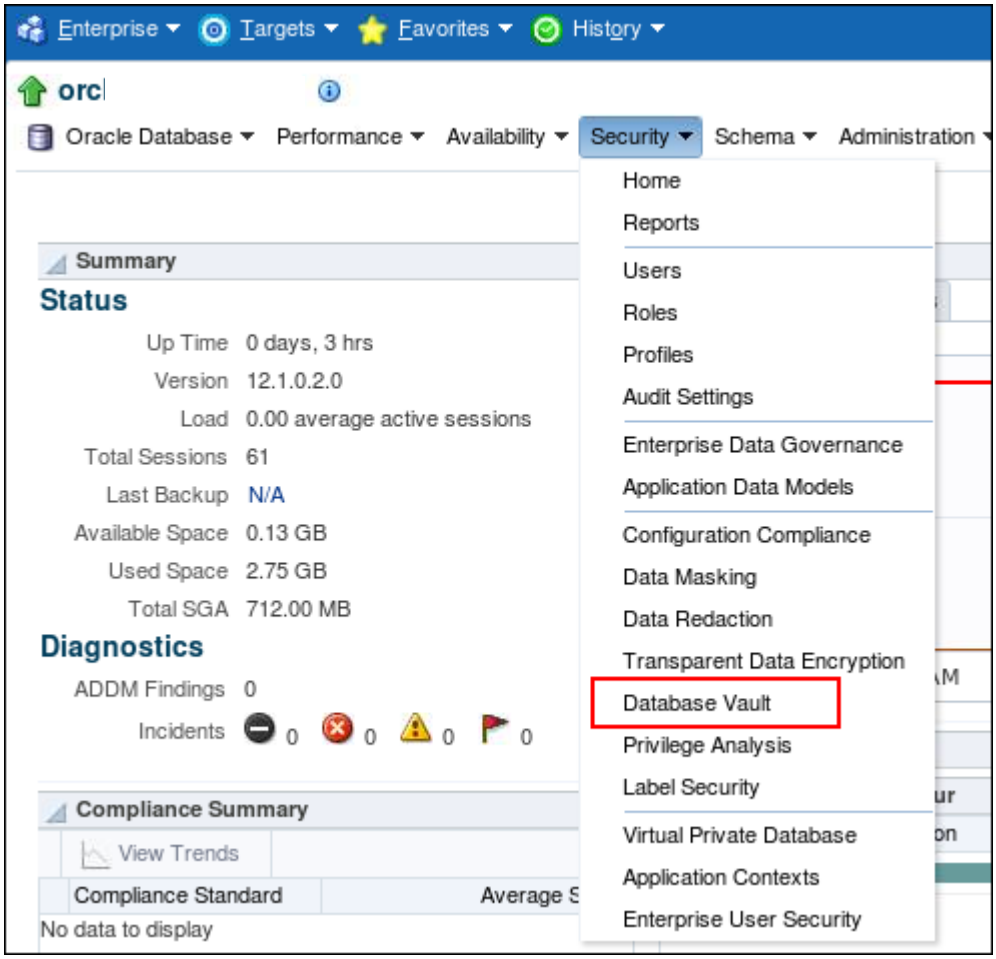
d.



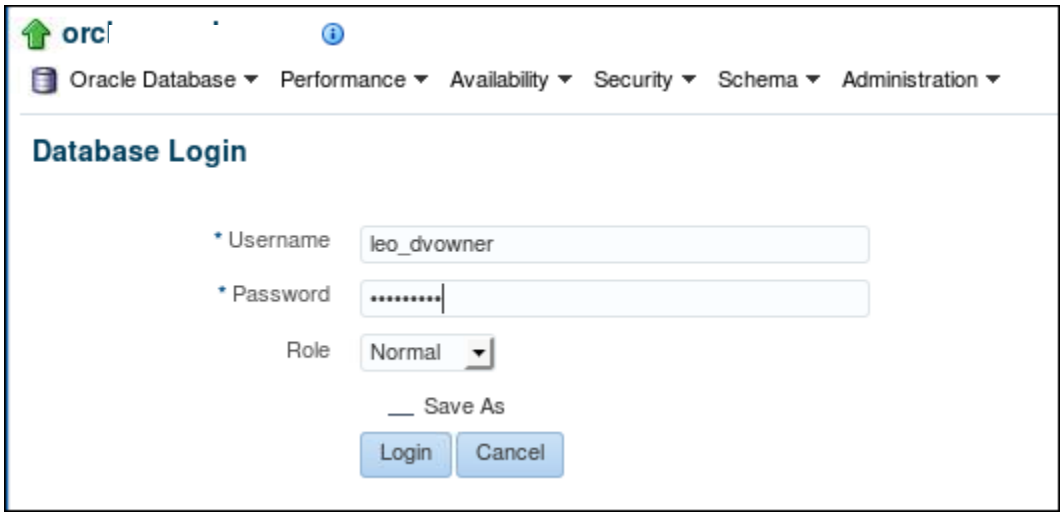
e.



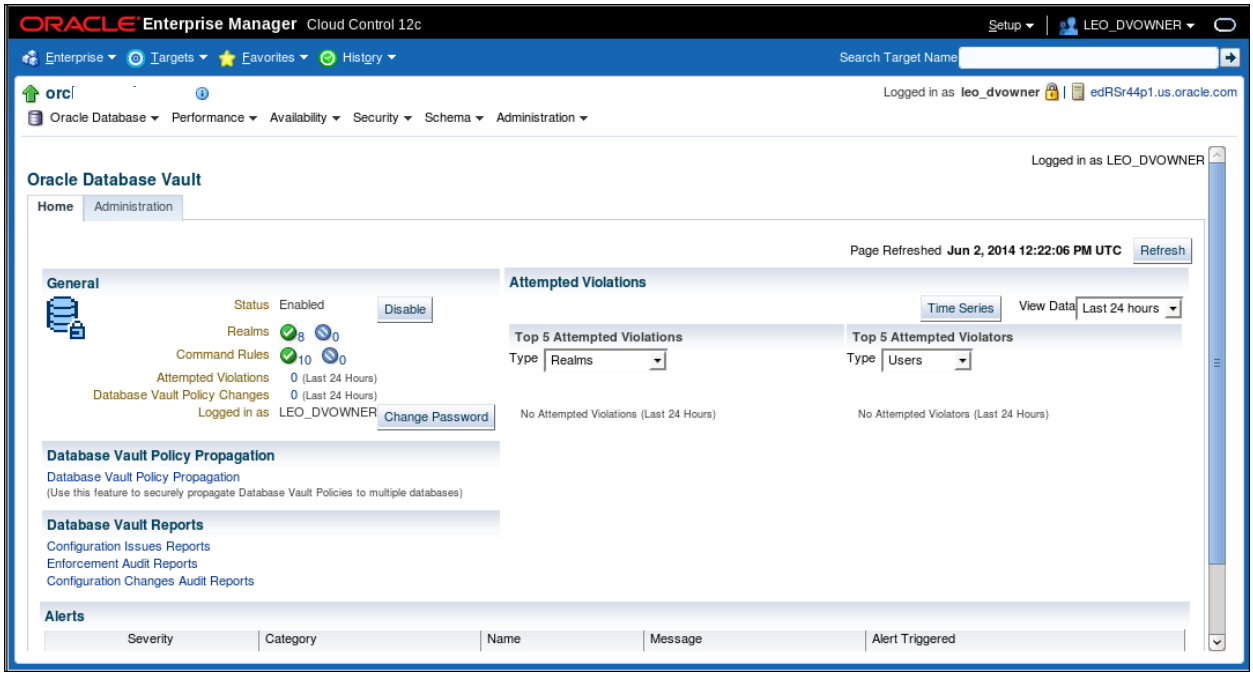
f.



g.



Success: The Oracle Database Vault Home page appears.



Practices for Lesson 4: Analyzing Privileges

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

In these practices, you will learn different ways to analyze privileges. You need this knowledge to analyze the privileges in your own organization and then to determine whether there are any users who might have more privileges than needed. To reduce the possibility of a security attack, only the truly required privileges should be active in production systems.

Choose the practice that is most useful to you. They are stand-alone practices (that is, no later practices depend on them).

Assumptions

Practices 3-1 and 3-2 were successfully completed:

- Database Vault is configured.
- Test users are created.

Practice 4-1: Analyzing Privileges Used by Any User

Overview

In this practice, you capture privileges used by users during a short period, generate the capture results, and compare between used and unused privileges to decide which privileges might need to be revoked.

Tasks

1. To create user accounts, log in to SQL* Plus as the BEA_DVACCTMGR user and create the PA_ADMIN (privilege analysis administrator) and the APP_USER user (end user).

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus bea_dvacctmgr
Enter password:

SQL> CREATE USER pa_admin IDENTIFIED BY oracle_4U;

User created.

SQL> CREATE USER app_user IDENTIFIED BY oracle_4U;

User created.

SQL> ALTER USER hr IDENTIFIED BY oracle_4U ACCOUNT UNLOCK;

User altered.

SQL>
```

2. As the DBA_PSMITH user, grant the CREATE SESSION and CAPTURE_ADMIN privileges to the PA_ADMIN user, who performs privilege analysis, and grant CREATE SESSION and SELECT ANY TABLE to the APP_USER user, who is the test user for this task.

```
SQL> connect DBA_PSMITH
Enter password:
Connected.
SQL> GRANT CREATE SESSION, CAPTURE_ADMIN TO pa_admin;

Grant succeeded.

SQL> GRANT CREATE SESSION, SELECT ANY TABLE TO app_user, hr;
```

```
Grant succeeded.

SQL> CREATE ROLE HR_MGR;

Role Created.

SQL> GRANT select, update,
           delete, insert ON hr.employees TO HR_MGR;

Grant succeeded.

SQL> grant HR_MGR TO app_user;

Grant succeeded.

SQL>
```

3. The PA_ADMIN user defines a capture of privileges used by all users.

```
exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
name           =>    'All_privs', -
description      =>  'All privs used', -
    type            =>    dbms_privilege_capture.g_database)
```

```
SQL> CONNECT pa_admin
Enter password:
Connected.
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE ( -
          name       =>    'All_privs', -
          description=>    'All privs used', -
           type       =>   dbms_privilege_capture.g_database)
> > >
PL/SQL procedure successfully completed.

SQL>
```

4. The PA_ADMIN user starts capturing the privileges while users are performing their daily work using privileges.
   a. Start the capture.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE ( -
          name => 'All_privs')
>
PL/SQL procedure successfully completed.
SQL>
```

   b. Connect as APP_USER and delete rows from the HR.EMPLOYEES table, and connect as HR and select rows from the SH.SALES table.

```
SQL> conn app_user
Enter password:
Connected.
SQL> delete hr.employees where salary<3000;

24 rows deleted.

SQL> rollback;

Rollback complete.

SQL> conn hr
Enter password:
Connected.
SQL> select * from sh.sales
     where amount_sold < 6.42
     and   cust_id= 6452;
  2    3
   PROD_ID    CUST_ID TIME_ID   CHANNEL_ID  PROMO_ID
QUANTITY_SOLD AMOUNT_SOLD
---------- ---------- --------- ---------- ----------
------------ -----------
       120       6452 29-SEP-00          2       999
            1         6.4
       120       6452 29-SEP-00          4       999
            1         6.4

SQL>
```

5. Stop the capture.

```
SQL> connect pa_admin
Enter password:
Connected.
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE ( -
          name      => 'All_privs')
>
PL/SQL procedure successfully completed.
SQL>
```

6. Generate the capture results. It may take a few seconds.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT ( -
          name => 'All_privs')
>
PL/SQL procedure successfully completed.

SQL>
```

7. Display the object privileges used during the capture period.

```
SQL> set pages 99
COL username FORMAT A10
COL object_owner FORMAT A12
COL object_name FORMAT A30
COL obj_priv FORMAT A25
SQL> SQL> SQL> SQL>
SQL> SELECT  username, object_owner, object_name, obj_priv
     FROM  dba_used_objprivs
     WHERE username IN ('APP_USER', 'HR')
     ORDER BY 1, 4;
  2    3    4
USERNAME   OBJECT_OWNER OBJECT_NAME                    OBJ_PRIV
---------- ------------ ------------------------------ ---------
----------------
APP_USER   HR           EMPLOYEES                      DELETE
APP_USER   SYS          DBMS_APPLICATION_INFO          EXECUTE
APP_USER   HR           EMPLOYEES                      SELECT
APP_USER   SYSTEM       PRODUCT_PRIVS                  SELECT
APP_USER   SYS          DUAL                           SELECT
APP_USER   SYS          DUAL                           SELECT
APP_USER   SYS          ORA$BASE
    USE
HR         SYS          DBMS_OUTPUT                    EXECUTE
HR         SYS          DBMS_APPLICATION_INFO          EXECUTE
HR         SYSTEM       PRODUCT_PRIVS                  SELECT
HR         SYS          DUAL                           SELECT
HR         SYS          DUAL                           SELECT
HR         SYS          ORA$BASE
    USE

13 rows selected.

SQL>
```

Note: Your row number and sequence may be slightly different, but you should see the `DELETE`, `EXECUTE`, `SELECT`, and `USE` privileges of the `APP_USER`.

8. Display the system privileges used.

```
SQL> COL  sys_priv form a16
SQL> SELECT username, sys_priv FROM dba_used_sysprivs
     WHERE  username IN ('APP_USER', 'HR');
  2
USERNAME   SYS_PRIV
---------- ----------------
HR         CREATE SESSION
HR         SELECT ANY TABLE
APP_USER   CREATE SESSION

SQL>
```

Note: The HR user uses an ANY privilege which grants more access than is needed.

9. Display the path of the privileges used if the privileges were granted to roles, and roles to users.

```
SQL> COL object_name FORMAT A10
COL path FORMAT A60
COL obj_priv FORMAT A10
SQL> SQL> SQL>
SQL> SELECT username, obj_priv, object_name, path
     FROM   dba_used_objprivs_path
     WHERE  username IN ('APP_USER', 'HR')
     AND    object_name IN ('SALES','EMPLOYEES');
  2    3    4
USERNAME   OBJ_PRIV   OBJECT_NAM
---------- ---------- ----------
PATH
--------------------------------
APP_USER   DELETE     EMPLOYEES
GRANT_PATH('APP_USER', 'HR_MGR')

APP_USER   SELECT     EMPLOYEES
GRANT_PATH('APP_USER', 'HR_MGR')

SQL>
```

10. APP_USER is granted select and delete privileges on the HR.EMPLOYEES table through HR_MGR role. He used the DELETE and SELECT privileges until now.
The unused privileges are visible in DBA_UNUSED_PRIVS view.

```
SQL> SELECT username, sys_priv, obj_priv, object_name, path
     FROM   dba_unused_privs
     WHERE  username='APP_USER';
```

```
 2    3
USERNAME   SYS_PRIV          OBJ_PRIV   OBJECT_NAM
---------- --------------- ---------- ----------
PATH
------------------------------------------------------------
APP_USER           UPDATE     EMPLOYEES
GRANT_PATH('APP_USER', 'HR_MGR')

APP_USER           INSERT     EMPLOYEES
GRANT_PATH('APP_USER', 'HR_MGR')

APP_USER   SELECT ANY TABLE
GRANT_PATH('APP_USER')

SQL>
```

11. You may have different rows displayed. Compare used and unused privileges to make a decision about a possible privilege revoking. This will not be decided in the context of the practice.
12. Display the definition of the capture. The `ENABLED` column ensures that the `All_privs` capture has been stopped.

```
SQL> COL name FORMAT A14
COL type FORMAT A12
COL enabled FORMAT A2
COL roles FORMAT A10
COL context FORMAT a10
SQL> SQL> SQL> SQL> SQL>
SQL> SELECT name, type, enabled,roles, context
     FROM dba_priv_captures;
  2
NAME           TYPE          EN ROLES      CONTEXT
-------------- ------------ -- ---------- ----------
All_privs      DATABASE     N

SQL>
```

13. Delete the capture, to remove all previous captured information from the views.
    a. Execute the procedure.

```
SQL> exec SYS.DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE ( -
          name => 'All_privs')
>
PL/SQL procedure successfully completed.

SQL>
```

b. Verify that no data from the `All_privs` capture is left.

```
SQL> SELECT username, sys_priv, obj_priv, object_name, path
     FROM   dba_unused_privs
     WHERE  username in ('APP_USER', 'HR');
  2    3
no rows selected

SQL>
```

c. As the `DBA_SMITH` user, revoke the `HR_MGR` role from `APP_USER` and drop the role, but keep the user who is still the test user for the next tasks.

```
SQL> connect DBA_PSMITH
Enter password:
Connected.
SQL> REVOKE hr_mgr FROM app_user;

Revoke succeeded.
SQL>
SQL> drop role hr_mgr;

Role dropped.

SQL>
```

Practice 4-2: Analyzing ANY Privilege Use in Context

Overview

In this practice, you analyze the use of the SELECT ANY TABLE system. The product documentation also shows this procedure step-by-step.

Assumptions

Practice 4-1 was successfully completed (the APP_USER and PA_ADMIN users exist.)

Tasks

1. Optionally, review the “Analyzing ANY Privilege Use” video unless your instructor just demonstrated the equivalent steps.
   a. Double-click the **demos** folder and then the **dv_priv_any** folder.
   b. In an Oracle classroom, double-click the HTML version (dv_priv_any.html) to start the video. In other environments, you may use the MP4 version of the same video.
2. As the PA_ADMIN user create a privilege analysis policy.

```
SQL> connect pa_admin
Enter password:
Connected.
SQL> BEGIN
 DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
  name           => 'ANY_priv_analysis_pol',
  description    => 'Analyzes system privilege use',
  type           => DBMS_PRIVILEGE_CAPTURE.G_CONTEXT,
  condition      => 'SYS_CONTEXT(''USERENV'',
''SESSION_USER'')=''APP_USER''');
END;
/
2    3    4    5    6    7    8
PL/SQL procedure successfully completed.

SQL>
```

3. Enable the newly created privilege analysis policy.

```
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE
('ANY_priv_analysis_pol')

PL/SQL procedure successfully completed.
SQL>
```

4. As the APP_USER user, create test data by using the SELECT ANY TABLE system privilege.

```
SQL> CONNECT app_user
Password:
Connected.
```

```
SQL> SELECT FIRST_NAME, LAST_NAME, SALARY
     FROM   HR.EMPLOYEES
     WHERE  salary > 12000
     ORDER BY salary DESC;
  2    3    4
FIRST_NAME           LAST_NAME                      SALARY
-------------------- ------------------------- ----------
Steven               King                            24000
Neena                Kochhar                         17000
Lex                  De Haan                         17000
John                 Russell                         14000
Karen                Partners                        13500
Michael              Hartstein                       13000
Nancy                Greenberg                       12008
Shelley              Higgins                         12008

8 rows selected.
SQL>
```

5. As the PA_ADMIN user, disable the privilege analysis policy.

```
SQL> CONNECT pa_admin
Password:
Connected.
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE
('ANY_priv_analysis_pol')

PL/SQL procedure successfully completed.
SQL>
```

6. Continue as the PA_ADMIN user; generate the privilege analysis report.

```
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT
('ANY_priv_analysis_pol')

PL/SQL procedure successfully completed.

SQL>
```

7. Query the DBA_USED_PRIVS view to review the data that you gathered by using the privilege analysis policy.

```
SQL> set pages 99
col username      format a10
col sys_priv      format a16
col object_owner  format a13
col object_name   format a23
SQL> SQL> SQL> SQL> SQL>
```

```
SQL> SELECT USERNAME, SYS_PRIV, OBJECT_OWNER, OBJECT_NAME
     FROM   DBA_USED_PRIVS
     WHERE  USERNAME = 'APP_USER';
   2   3
USERNAME   SYS_PRIV          OBJECT_OWNER  OBJECT_NAME
---------- --------------- ------------ ----------------------
-
APP_USER   CREATE SESSION
APP_USER                   SYS           ORA$BASE
APP_USER                   SYS           DBMS_APPLICATION_INFO
APP_USER   SELECT ANY TABLE HR           EMPLOYEES
APP_USER                   SYS           DUAL
APP_USER                   SYS           DUAL
APP_USER                   SYSTEM        PRODUCT_PRIVS

7 rows selected.

SQL>
```

Question: Does the APP_USER use an ANY system privilege?

Answer: Yes.

8. As the PA_ADMIN user, remove the privilege capture.

```
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE
('ANY_priv_analysis_pol')

PL/SQL procedure successfully completed.

SQL>
```

9. As the BEA_DVACCTMGR user, drop the PA_ADMIN and the APP_USER users to make this practice repeatable.

```
SQL> connect bea_dvacctmgr
Password:
Connected.
SQL> DROP USER pa_admin CASCADE;
User dropped.
SQL> DROP USER app_user CASCADE;

User dropped.
SQL>
```

Note: You can keep your SQL session open for the next practice.

Practice 4-3: Analyzing Role-Based Privileges

Overview

In this practice, you analyze the system and object privileges use of a user who has been granted the DBA role and who performs database tuning operations.

Tasks

1. Optionally, review the "Analyzing Privilege Use by a User Who Has the DBA Role" video unless your instructor just demonstrated the equivalent steps.
   a. Double-click the **demos** folder and then the **dv_priv_dba** folder.
   a. In an Oracle classroom, double-click the HTML version (dv_priv_dba.html) to start the video. In other environments you may use the MP4 version of the same video.
2. Logged in to SQL* Plus as the BEA_DVACCTMGR user, create the PA_ADMIN and the TJONES users.

```
SQL> CREATE USER pa_admin IDENTIFIED BY oracle_4U;
User created.

SQL> CREATE USER tjones IDENTIFIED BY oracle_4U;

User created.

SQL>
```

3. As the DBA_PSMITH user, grant CREATE SESSION and DBA to TJONES, who is the test user for this task.

```
SQL> connect dba_psmith
Connected.
SQL> GRANT CREATE SESSION, CAPTURE_ADMIN TO pa_admin;
Grant succeeded.

SQL> GRANT CREATE SESSION, DBA TO tjones;
Grant succeeded.

SQL>
```

4. As the PA_ADMIN user, create a privilege analysis policy.

```
SQL> CONNECT pa_admin
Password:
Connected.
SQL> BEGIN
DBMS_PRIVILEGE_CAPTURE.CREATE_CAPTURE(
  name              => 'dba_role_analysis_pol',
  description       => 'Analyzes DBA role use',
  type              => DBMS_PRIVILEGE_CAPTURE.G_ROLE,
  roles             =>  role_name_list('DBA'));
```

```
END;
/
  2    3    4    5    6    7    8   9

PL/SQL procedure successfully completed.
SQL>
```

5. Enable the privilege analysis policy.

```
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.ENABLE_CAPTURE
('dba_role_analysis_pol')

PL/SQL procedure successfully completed.
SQL>
```

6. As the TJONES user, perform a database tuning operation following the commands in the code box.

```
SQL> CONNECT tjones
Enter password:
Connected.
SQL> @$ORACLE_HOME/rdbms/admin/utlxplan.sql

Table created.

SQL> EXPLAIN PLAN
 SET STATEMENT_ID = 'Raise in Tokyo'
 INTO PLAN_TABLE
 FOR UPDATE HR.EMPLOYEES
 SET SALARY = SALARY * 1.10
 WHERE DEPARTMENT_ID =
  (SELECT DEPARTMENT_ID FROM HR.DEPARTMENTS WHERE LOCATION_ID =
110);
  2    3    4    5    6    7
Explained.

SQL> @$ORACLE_HOME/rdbms/admin/utlchain.sql
Table created.

SQL> ANALYZE TABLE HR.EMPLOYEES LIST CHAINED ROWS INTO
CHAINED_ROWS;

Table analyzed.
SQL>
```

7. As the PA_ADMIN user, disable the privilege analysis.

```
SQL> connect pa_admin
Enter password:
Connected.
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.DISABLE_CAPTURE
('dba_role_analysis_pol')

PL/SQL procedure successfully completed.
SQL>
```

8. Generate privilege analysis reports.

```
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.GENERATE_RESULT
('dba_role_analysis_pol')

PL/SQL procedure successfully completed.
SQL>
```

9. Query the DBA_USED_SYSPRIVS_PATH view to review the system privileges and roles active for the TJONES user.

```
SQL>
set pages 99
col username     format a8
col sys_priv     format a18
col used_role    format a26
col path         format a150
col obj_priv     format a10
col object_owner format a10
col object_name  format a10
col object_type  format a10
SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL>
SQL> SELECT USERNAME, SYS_PRIV, USED_ROLE, PATH
     FROM   DBA_USED_SYSPRIVS_PATH
     WHERE  USERNAME = 'TJONES'
     ORDER BY USERNAME, SYS_PRIV, USED_ROLE;
  2    3    4
USERNAME SYS_PRIV           USED_ROLE
-------- ------------------ --------------------------
PATH
------------------------------------------------------------------
---
TJONES    ANALYZE ANY         IMP_FULL_DATABASE
GRANT_PATH('TJONES', 'DBA', 'DATAPUMP_IMP_FULL_DATABASE',
'IMP_FULL_DATABASE')

TJONES    ANALYZE ANY         IMP_FULL_DATABASE
```

```
GRANT_PATH('TJONES', 'DBA', 'IMP_FULL_DATABASE')

TJONES    CREATE SESSION     EM_EXPRESS_BASIC
GRANT_PATH('TJONES', 'DBA', 'EM_EXPRESS_ALL',
'EM_EXPRESS_BASIC')

TJONES    CREATE TABLE       DATAPUMP_EXP_FULL_DATABASE
GRANT_PATH('TJONES', 'DBA', 'DATAPUMP_EXP_FULL_DATABASE')

TJONES    CREATE TABLE       DATAPUMP_EXP_FULL_DATABASE
GRANT_PATH('TJONES', 'DBA', 'DATAPUMP_EXP_FULL_DATABASE',
'EXP_FULL_DATABASE')

TJONES    SELECT ANY TABLE   OLAP_DBA
GRANT_PATH('TJONES', 'DBA', 'OLAP_DBA')

TJONES    SELECT ANY TABLE   OLAP_DBA
GRANT_PATH('TJONES', 'DBA', 'OLAP_DBA')

TJONES    UPDATE ANY TABLE   OLAP_DBA
GRANT_PATH('TJONES', 'DBA', 'OLAP_DBA')

8 rows selected.
SQL>
```

10. Query the DBA_USED_OBJPRIVS view to review the object privileges active for the TJONES user.

```
SQL>
col username    format a6
col used_role   format a9
col object_name format a22
col object_type format a12
col obj_priv format a8
col owner format a6
col type format a7
SQL> SQL> SQL> SQL> SQL> SQL> SQL>
SQL> SELECT USERNAME, OBJ_PRIV, USED_ROLE,
            OBJECT_OWNER "OWNER", OBJECT_NAME,
            OBJECT_TYPE "TYPE"
     FROM  DBA_USED_OBJPRIVS
     WHERE USERNAME = 'TJONES'
     ORDER BY 1, 2, 3, 4, 5, 6;
  2    3    4   5   6
no rows selected

SQL>
```

Note: No object privilege is active.

11. Query the DBA_UNUSED_SYSPRIVS_PATH view to count the system privileges that were given to the TJONES user through the DBA role but that were not used.

```
SQL> set lines 100
col username    format a9
col sys_priv    format a40
SQL> SQL> SQL>
SQL> SELECT count(distinct sys_priv)
     FROM DBA_UNUSED_SYSPRIVS_PATH
     WHERE capture = 'dba_role_analysis_pol'
     ORDER BY SYS_PRIV;
2    3    4
COUNT(DISTINCTSYS_PRIV)
-----------------------
                    205
SQL>
```

12. Drop the privilege analysis capture.

```
SQL> EXEC DBMS_PRIVILEGE_CAPTURE.DROP_CAPTURE
('dba_role_analysis_pol')

PL/SQL procedure successfully completed.
SQL>
```

13. As the BEA_DVACCTMGR user, drop the PA_ADMIN and TJONES users to make this practice repeatable.

```
SQL> connect bea_dvacctmgr
Password:
Connected.
SQL> DROP USER pa_admin CASCADE;

User dropped.
SQL>
SQL> DROP USER tjones CASCADE;

User dropped.
SQL> EXIT
$
```

Practices for Lesson 5: Configuring Realms

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In these practices, you use a realm to protect an application schema. Then you authorize only the application's DBA to access the data. This is the simplest, most basic and effective functionality provided by Database Vault. The result of Practice 5-1 is used in later practices.

You can choose among the following practices according to what is most relevant for your organization. If time permits, complete all:

- In the second practice, you also use a realm to prevent unauthorized granting of a role.
- In the third practice, you explore the differences between regular and mandatory realms.

Assumptions

Practices 3-1, 3-2, and 3-4 were successfully completed:

- Database Vault is configured.
- Test users are created.
- The `leo_dvowner` user is configured to have access to Database Vault via EM Cloud Control.

Practice 5-1: Using Realms to Protect a Schema

Overview

In this **mandatory** practice, you protect the **HR** schema by using a realm.

Tasks

1. To illustrate a security issue that you have in your environment, log in to SQL*Plus as the **AHUNOLD** user and create an **HR.EMPLOYEES_COPY** table that is a copy of the **HR.EMPLOYEES** table.

   Question: Why is **AHUNOLD** able to create a table in the **HR** schema?

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus ahunold
Enter password:
SQL> create table hr.employees_copy as
     select * from hr.employees;
  2
Table created.


SQL>
```

   Answer: **AHUNOLD** is able to create a table in the **HR** schema because **AHUNOLD** is granted the `DBA` role, which has the `CREATE ANY TABLE` system privilege.

2. The **HR** application DBA, `BERNST`, notices the new table and decides to protect the schema from other DBAs. He or she requests that the security manager protect the schema with a realm.

   Log in to Cloud Control as the `leo_dvowner` user and create a realm called `HR Schema` that prohibits all users from accessing any objects in the `HR schema`. The exception: Grant realm access as `Participant` to the `BERNST` user.

   If you want to confirm how to log in and navigate to the Oracle Database Vault home page, review Practice 3-4, steps 3d to 3f.

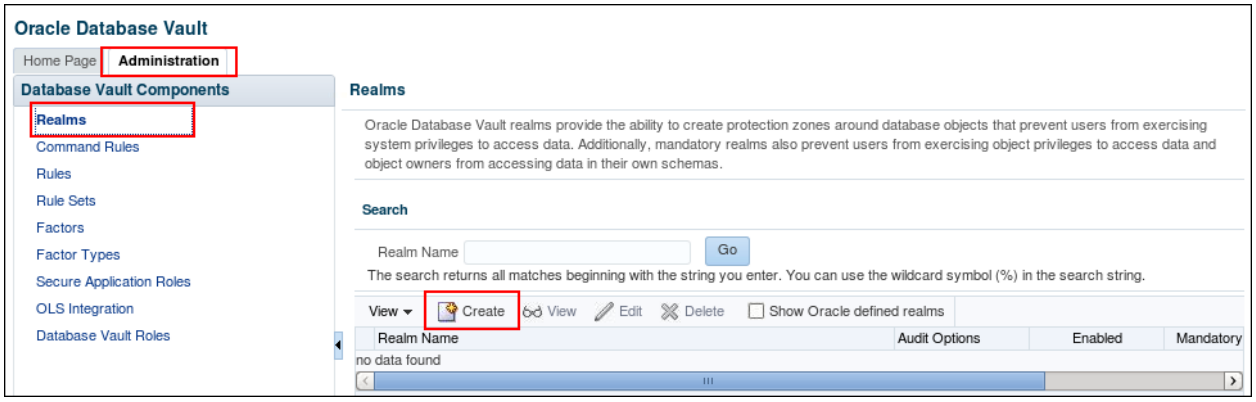
   For further assistance, see the screenshots below.

   a. Navigate to Administration (tabbed page) > Realms > Create (button).
   b. On the Create Realm: General page, enter `HR_Schema` as **Name**, optionally provide a description, accept Status **Enabled** and **Audit on Failure**, and then click **Next**.
   c. Click **Add**, then enter `HR` in the Add Secured Object window, accept `%` for Object Name and Object Type, and then click **OK**.
   d. Note that all objects in the HR schema are “Realm Secured Objects.” Click **Next**.
   e. On the Create Realm: Realm Authorizations page, click **Add**. In the Add Authorization window, enter `BERNST` as **Realm Authorization Grantee**, accept **Participant** as the **Realm Authorization Type** and click **OK**.
   f. Review `BERNST` as realm participant and click **Next**.

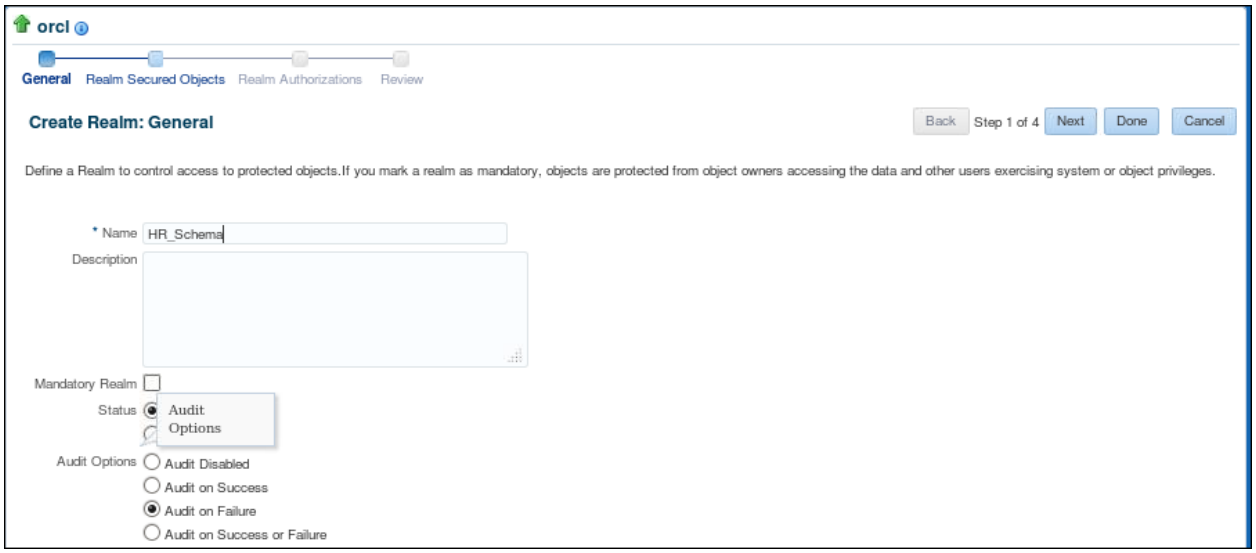
g. Review your `HR_Schema` definition including the PL/SQL code at the bottom of the page. (*This code is especially important if you do not have Cloud Control as your job aid and need to perform the equivalent task with a command line tool.*) Then click **Finish.**

h. You should receive a success message.

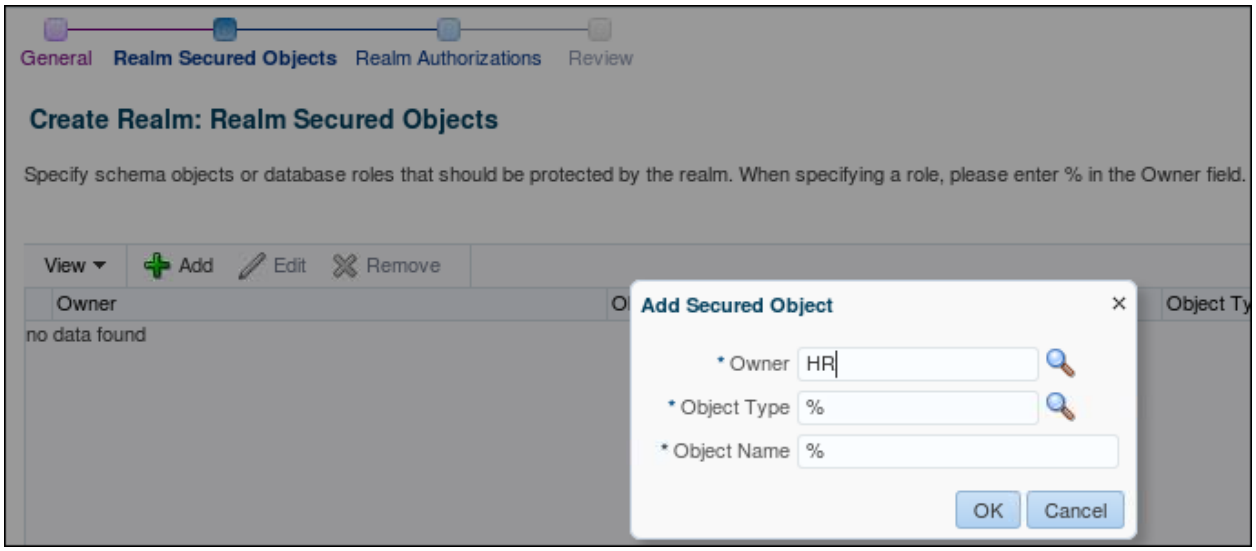
a.



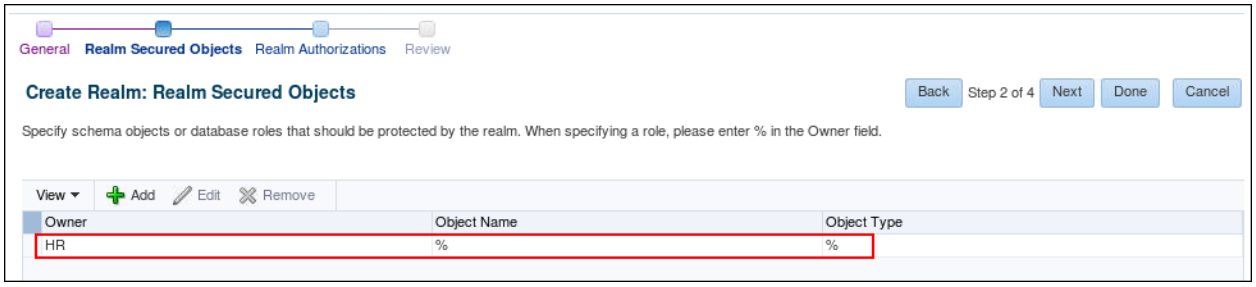
b.



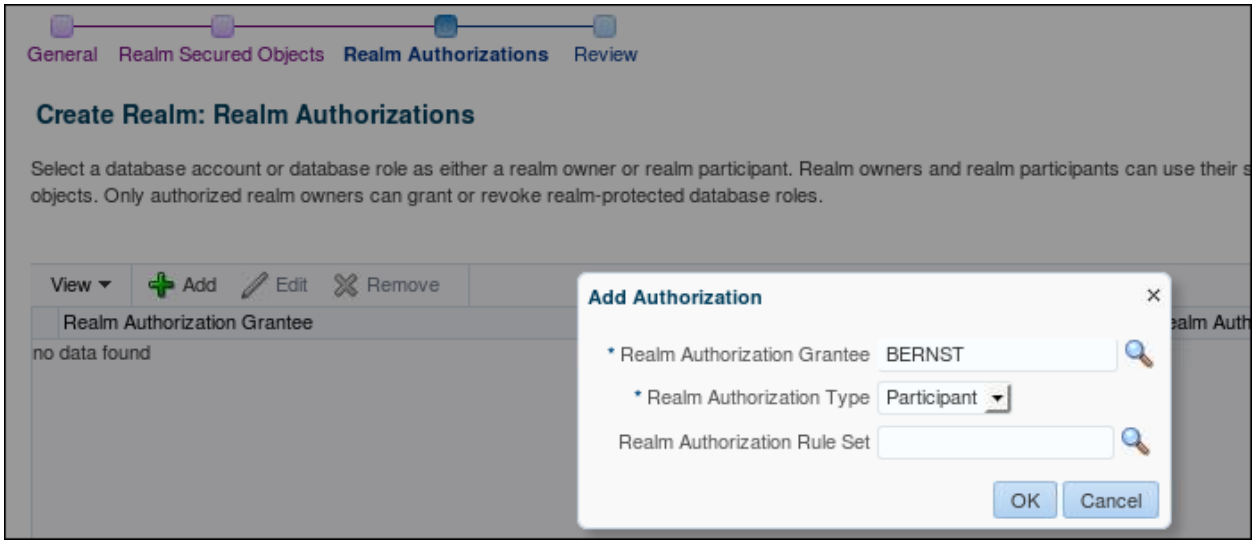
c.



d.

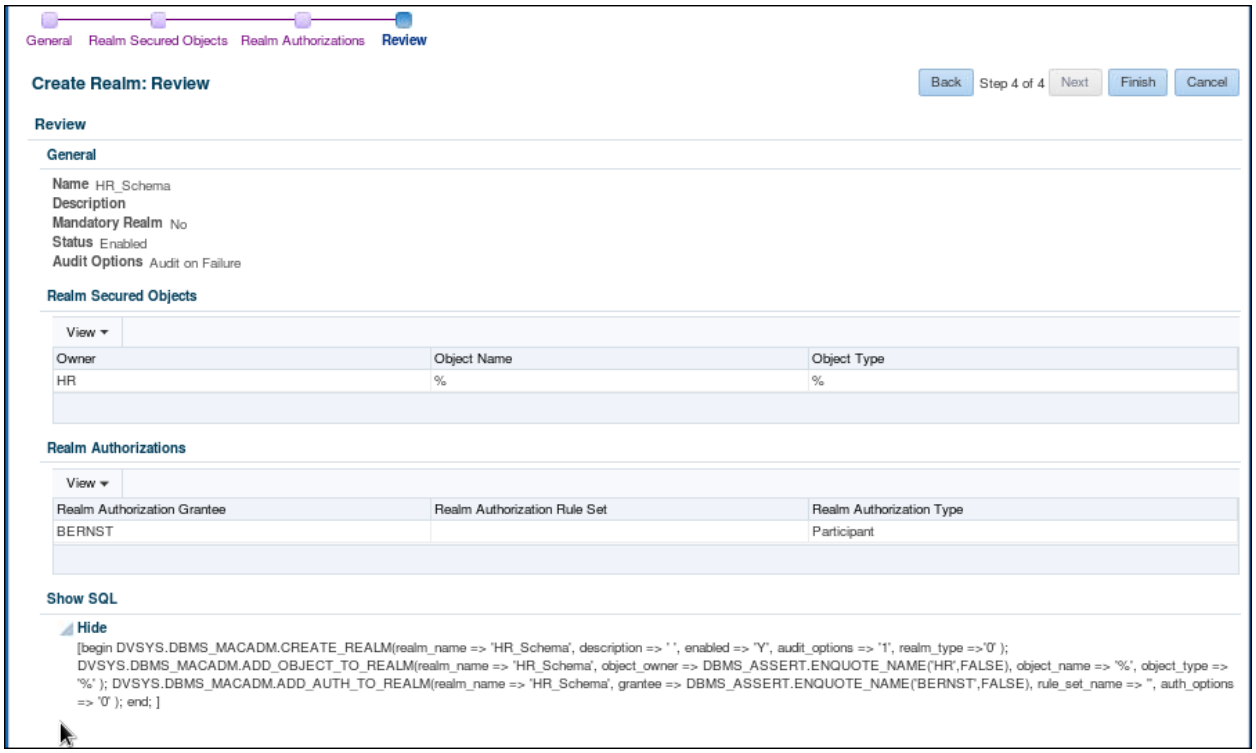


e.

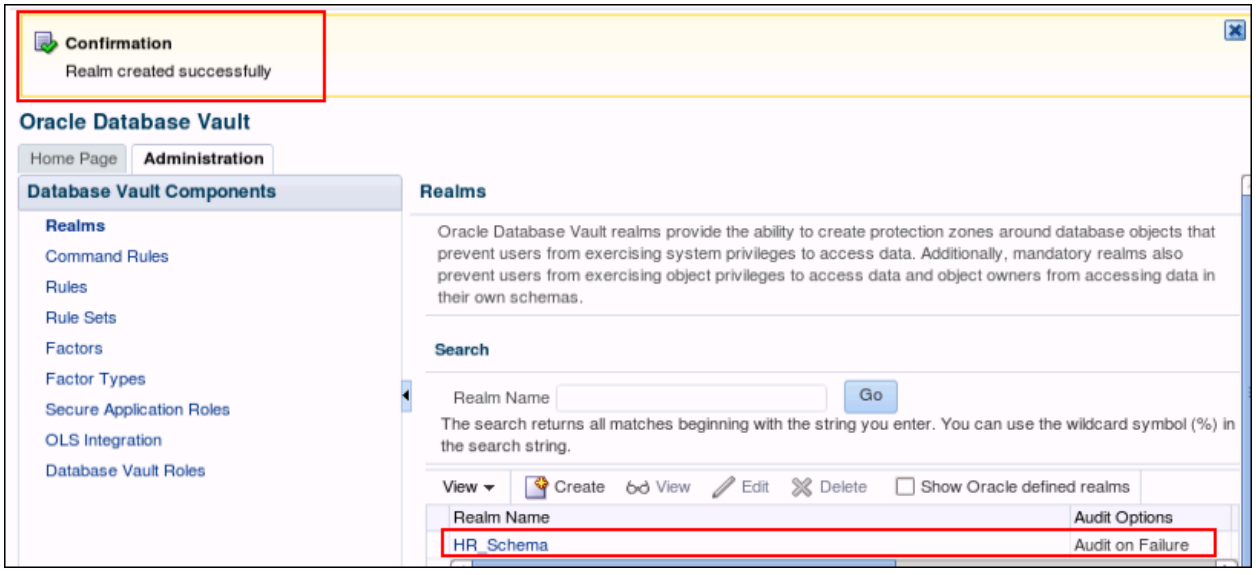


f.

g.



h.



Note: *You will return to the Cloud Control session.*

3. Logged in to SQL*Plus as the AHUNOLD user, attempt to drop the HR.EMPLOYEES_COPY table.

   Question: Why can the AHUNOLD user not drop his or her own table?

```
SQL> show user
USER is "AHUNOLD"
SQL> drop table hr.employees_copy;
drop table hr.employees_copy
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on HR.EMPLOYEES_COPY


SQL>
```

   Answer: The AHUNOLD user cannot drop his or her own table because the entire HR schema is now protected by the HR schema realm and the AHUNOLD user is not a realm participant.

4. Connect as the BERNST user and attempt to drop the HR.EMPLOYEES_COPY table.

   Question: Why does it succeed?

```
SQL> connect bernst
Enter password:
Connected.
SQL> drop table hr.employees_copy;

Table dropped.
SQL>
```

   Answer: The BERSNT user is a realm participant and has been granted the HR_DBA role (in Practice 3).

5. Now test using the HR user. Attempt to update the HR.DEPARTMENTS table, setting department_name to HR where department_id =40.

   Question: Why is the update allowed?

```
SQL> connect hr
Enter password:
Connected.
SQL>
SQL> update hr.departments
 set department_name = 'HR'
 where department_id = 40;
  2    3
1 row updated.

SQL>
```

Answer: The HR schema owner can update within his or her schema because this is a regular realm; with a **mandatory** realm they could not perform this update.

6. Roll back the update.

```
SQL> rollback;

Rollback complete.

SQL>
```

7. Test the creation of a table in the HR schema as the HR user.

Question: Why is HR not able to create a table in his or her own schema?

```
SQL> create table hr.x (a number);
create table hr.x (a number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on HR.X

SQL>
```

Answer: He or she cannot create objects because he or she is neither a realm participant nor a realm owner.

8. Check the USER_TAB_PRIVS_RECD view to see whether HR has any object-level privileges.

```
SQL> select * from user_tab_privs_recd;

OWNER
------------------------------------------------------------------
TABLE_NAME
------------------------------------------------------------------
GRANTOR
------------------------------------------------------------------
PRIVILEGE                       GRA HIE COM TYPE
--------------------------------------- --- --- --- ----------
SYS
DBMS_STATS
SYS
EXECUTE                         NO  NO  NO  PACKAGE

SQL> exit
$
```

Note: The SYS user granted the HR user the EXECUTE privilege on the DBMS_STATS package.

Practice 5-2: Using Realms to Protect Roles

Overview

In this optional practice, you protect a role by using a realm, and then demonstrate that if a role is protected by a realm, only members of the realm are able to grant that role.

Tasks

1. Log in as the **BERNST** user and create a new role called **HR_MGR**.

```
$ sqlplus bernst
Enter password:
SQL>
SQL> create role hr_mgr;

Role created.
SQL>
```

2. If you are no more logged in to Cloud Control as the `leo_dvowner` user, first log in to Cloud Control as the `leo_dvowner` user. Then create an `HR_Manager` realm, add the `HR_MGR` role to it, and grant realm access to the `SMAVRIS` user.

   For further assistance, see the screenshots below.

   a. Navigate to Administration (tabbed page) > Realms > Create (button).

   b. On the Create Realm: General page, enter `HR_Manager` as **Name**, optionally provide a description, accept Status **Enabled** and **Audit on Failure**, and then click **Next**.

   c. Click **Add**, then enter or select `ANONYMOUS` as **Owner**, enter `ROLE` as the **Object Type** and `HR_MGR` as **Object Name**, and then click **OK.**

   d. Review your realm secured objects and click **Next**.

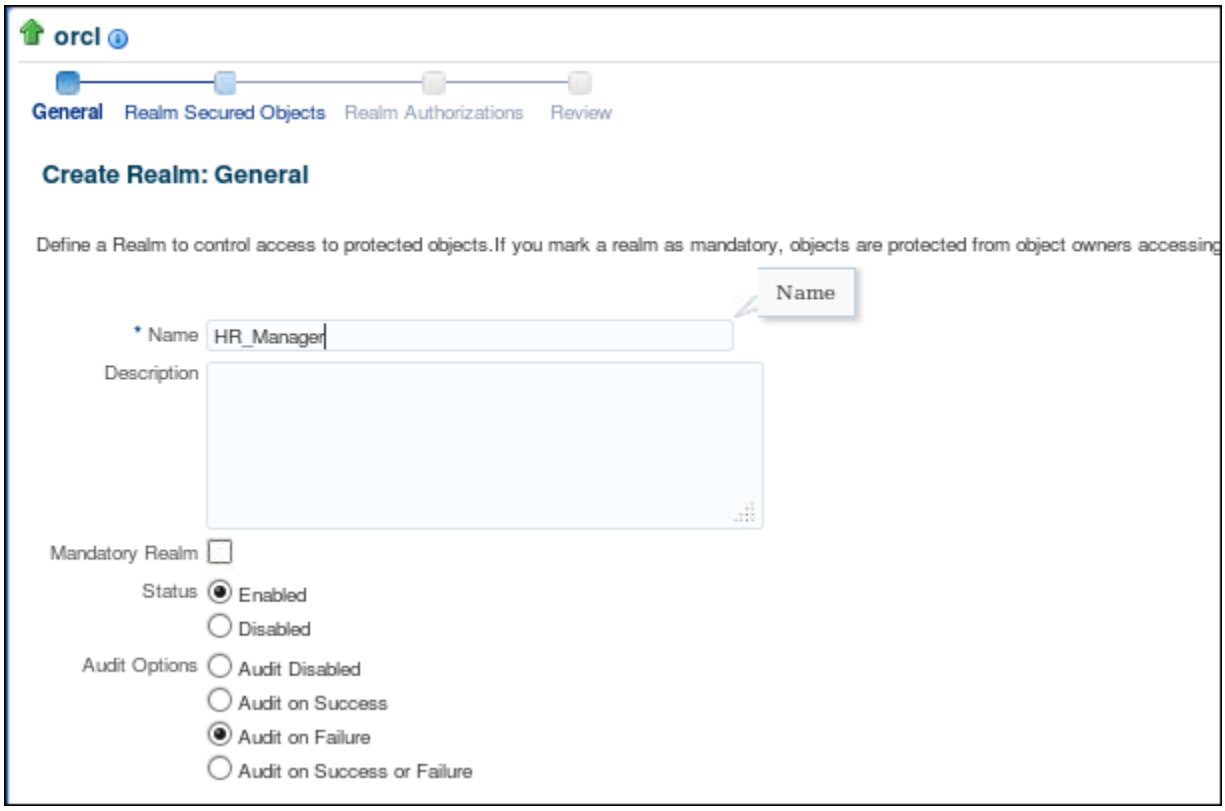
   e. On the Create Realm: Realm Authorizations page, click **Add**. In the Add Authorization window, enter or select `SMAVRIS` as **Realm Authorization Grantee**, accept **Participant** as the **Realm Authorization Type**, and click **OK**.

   f. Review your realm authorizations and click **Next**.

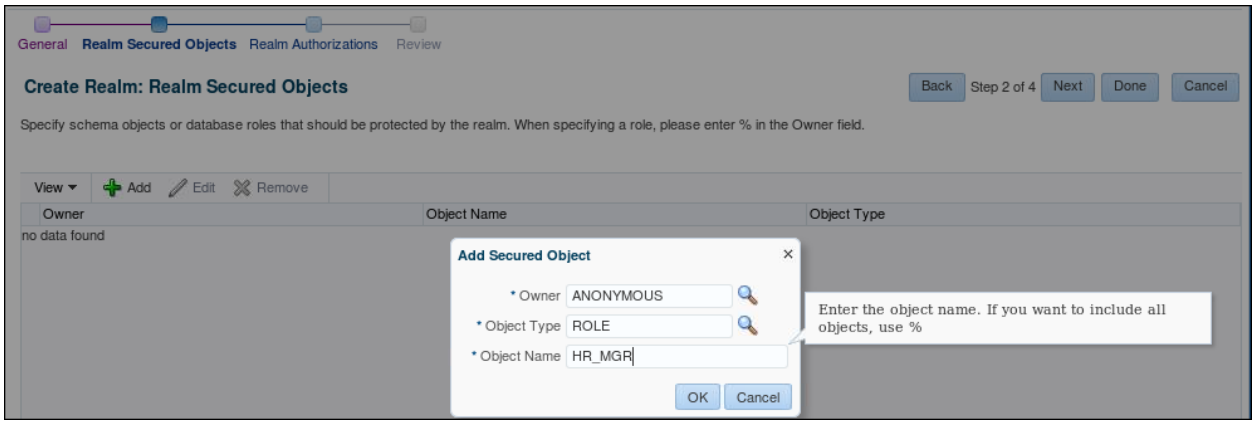
   g. Review your `HR Manager` definition including the SQL code at the bottom of the page. Then click **Finish.**

   h. You should receive a success message.

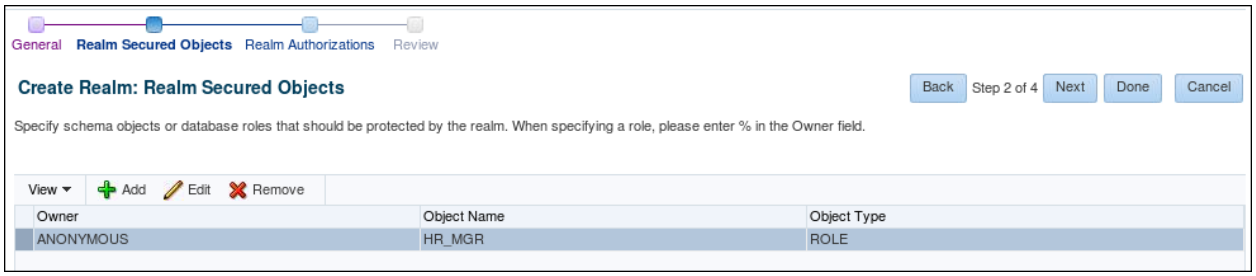
b.



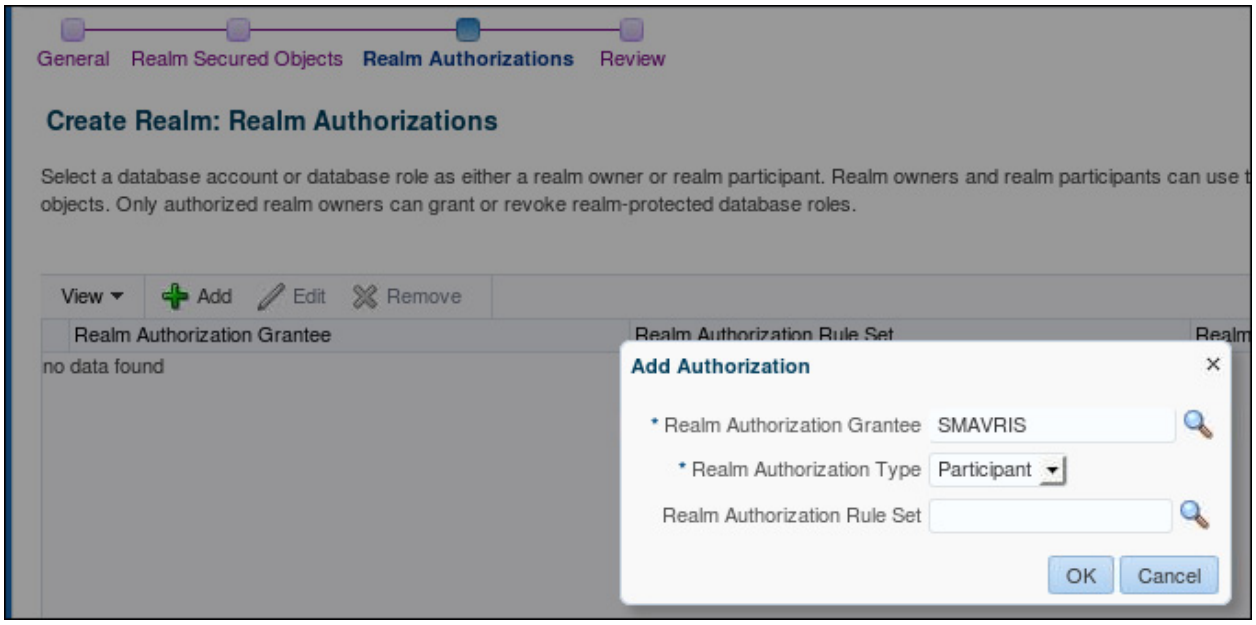
c.



d.

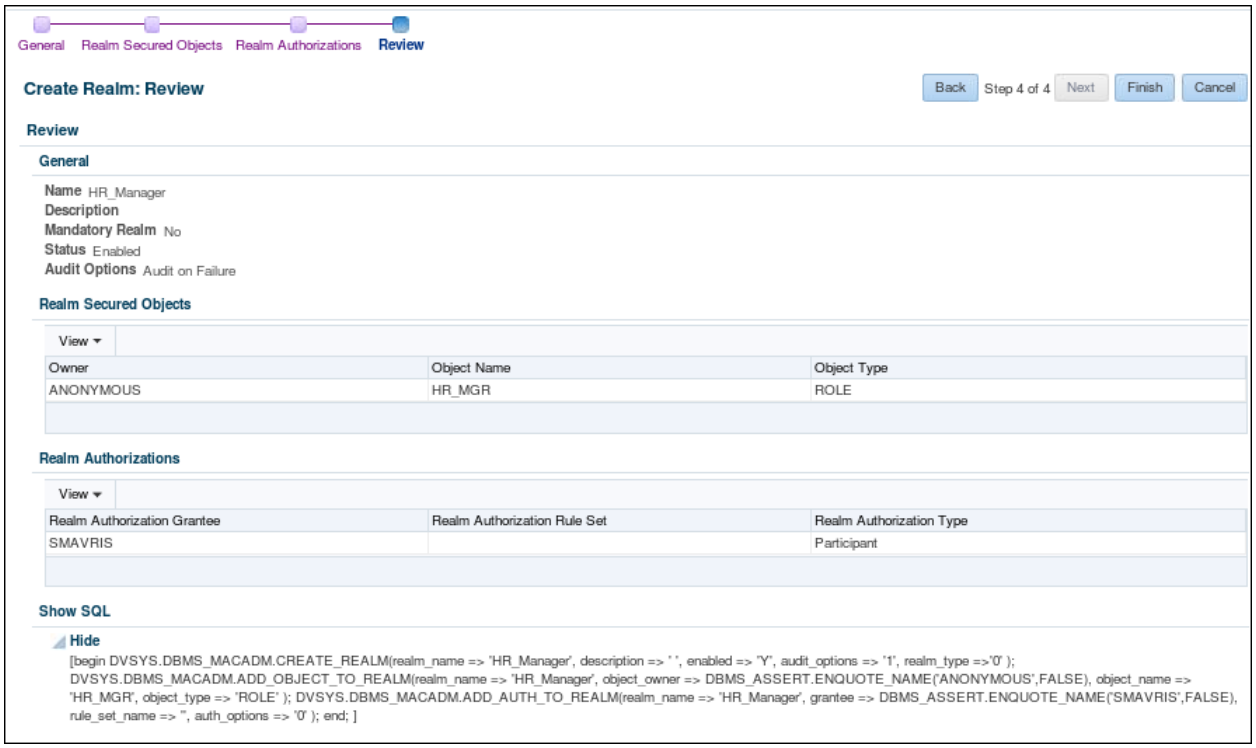


e.

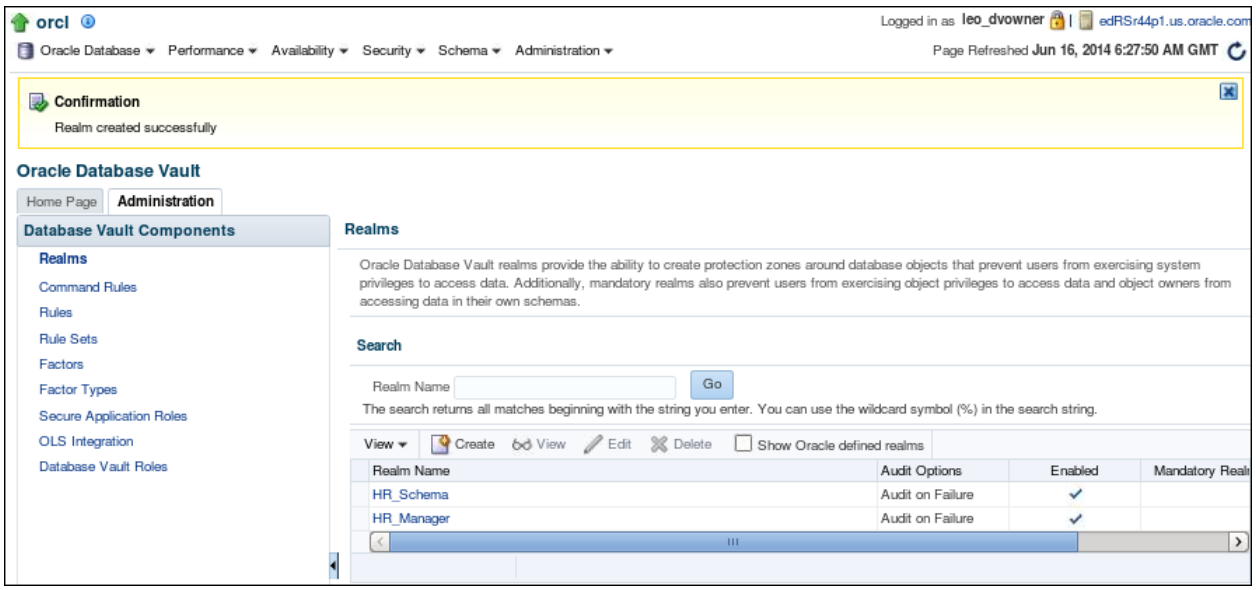


f.

g.



h.



3. Return to the SQL*Plus session of `BERNST` and attempt to grant the `HR_MGR` role to `KPARTNER`.

   Question: What is the result?

   Tip: If you are unsure about who is logged in to a SQL*Plus session, use the `show user` SQL*Plus command.

```
SQL> show user
USER is "BERNST"
```

```
SQL>
SQL> grant hr_mgr to kpartner;
grant hr_mgr to kpartner
*
ERROR at line 1:
ORA-47410: Realm violation for GRANT on HR_MGR


SQL>
```

Answer: Although `BERNST` created the role, he or she cannot grant the role to any other users. This is because the role is protected by the `HR Manager` realm and `BERNST` has no authorization on that realm.

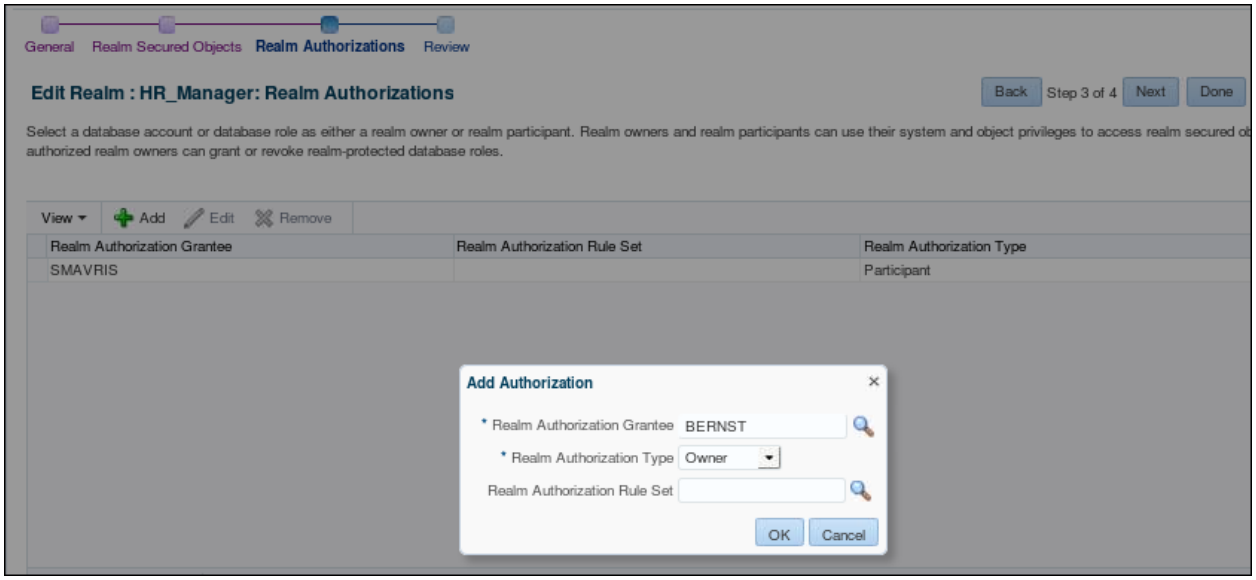
4. Connect as the `SMAVRIS` user and attempt to grant the `HR_MGR` role to `KPARTNER`.

   Question: What is the result?

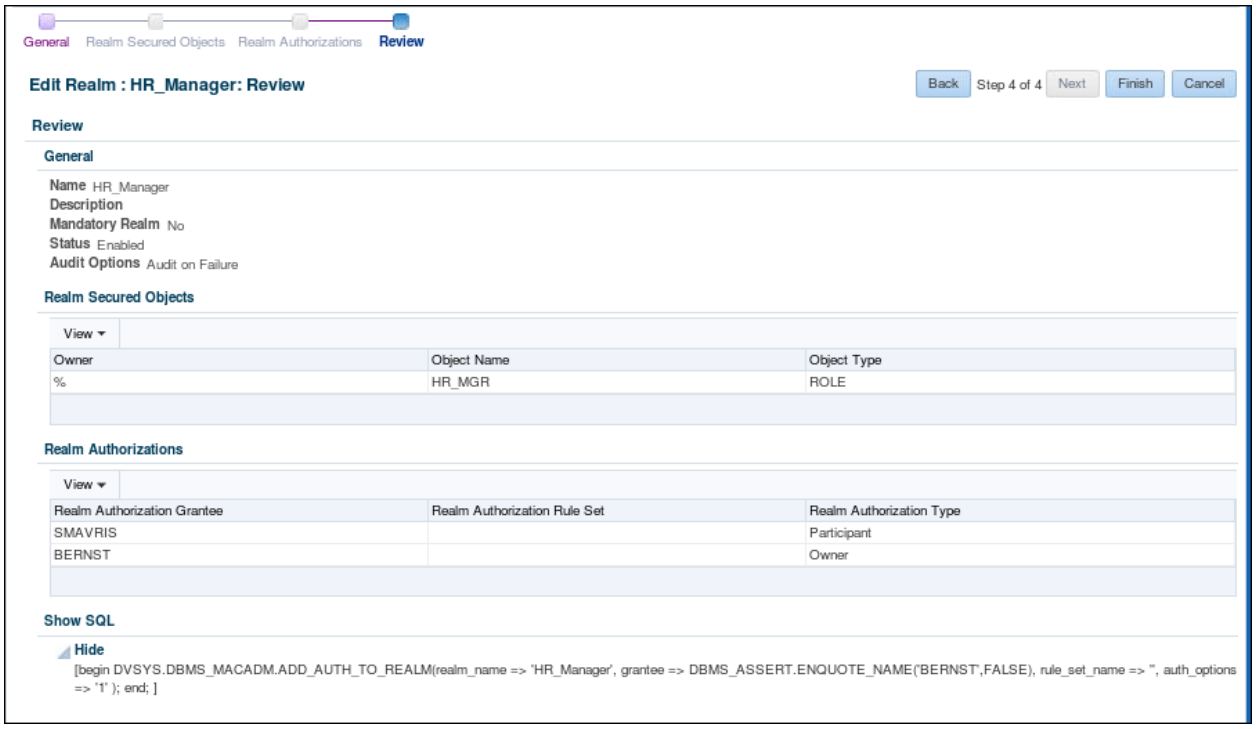
```
SQL> connect smavris
Enter password:
Connected.
SQL> grant hr_mgr to kpartner;
grant hr_mgr to kpartner
*
ERROR at line 1:
ORA-47410: Realm violation for GRANT on HR_MGR

SQL>
```

   Answer: Although `SMAVRIS` is a participant of the `HR_Manager` realm, he or she cannot grant the `HR_MGR` role.

5. As `leo_dvowner`, make the `BERNST` user an **Owner** in the `HR_Manager` realm. (If you are looking for navigational aid, review the steps under task 2.)



6. Review your modification and click Finish.



7. Connect as `BERNST` and reattempt the grant as described above.

   Question: What is the result now?

```
SQL> connect bernst
Enter password:
Connected.
SQL> grant hr_mgr to kpartner;

Grant succeeded.

SQL> EXIT
$
```

Answer: Success! As an owner in the realm, `BERNST` can grant the realm-protected `HR_MGR` role. It does not matter whether the grantee `KPARTNER` is in the realm or not.

Practice 5-3: Using Regular and Mandatory Realms

Overview

In this optional practice, you learn about the difference in protections levels between regular and mandatory realms in Database Vault. Mandatory realms are new to Database Vault with Oracle Database 12*c*.

In the previous practices, you saw how to create a realm by using Cloud Control. Now, for learning purposes, the tasks are demonstrated using the command line. You could use Cloud Control to accomplish the same thing without having to use the command line.

Tasks

1. Optionally, review the "Using Regular and Mandatory Realms in Oracle Database Vault" video directory unless your instructor just demonstrated the equivalent steps.
   a. Double-click the **demos** folder and then the **dv02_realm** folder.
   b. In an Oracle classroom, double-click the HTML version (`dv02_realm.html`) to start the video. In other environments, you may use the MP4 version of the same video.
2. Start a new SQL*Plus session as the `leo_dvowner` and create the `HR_Application` regular realm.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus leo_dvowner
Enter password:
SQL>
SQL> BEGIN
 DVSYS.DBMS_MACADM.CREATE_REALM(
   realm_name    => 'HR_Application',
   description   => 'Realm to protect the HR application',
   enabled       => DBMS_MACUTL.G_YES,
   audit_options => DBMS_MACUTL.G_REALM_AUDIT_FAIL +
DBMS_MACUTL.G_REALM_AUDIT_SUCCESS,
   realm_type    => 0);
 END;
/
  2    3    4    5    6    7    8    9

PL/SQL procedure successfully completed.

SQL>
```

Note: Realm type `0` means that the realm is a regular one; realm type `1` means that it is a mandatory realm.

3. Add the HR.EMPLOYEES table to the HR_Application realm.

```
SQL> BEGIN
 DVSYS.DBMS_MACADM.ADD_OBJECT_TO_REALM(
  realm_name   => 'HR_Application',
  object_owner => 'HR',
  object_name  => 'EMPLOYEES',
  object_type  => 'TABLE');
END;
/
  2    3    4    5    6    7    8
PL/SQL procedure successfully completed.

SQL>
```

4. View the protected table by querying from DVSYS.DBA_DV_REALM and DVSYS.DBA_DV_REALM_OBJECT.

```
SQL> col name   format a20
col object_name format a20
SQL> SQL>
SQL> SELECT r.name, o.object_name, r.realm_type
     FROM   dvsys.dba_dv_realm r,dvsys.dba_dv_realm_object o
     WHERE  r.name = o.realm_name
     AND   o.object_name = 'EMPLOYEES';
   2    3    4
NAME                 OBJECT_NAME          REALM_TYPE
-------------------- -------------------- -----------------
HR_Application       EMPLOYEES            REGULAR

SQL>
```

5. In another terminal window, connect as the HR user and test access to the EMPLOYEES table by querying one row. Your resulting row may be different.

```
$ . oraenv
ORACLE_SID = [oracle] ? orcl
$ sqlplus hr
Enter password:
Connected.
SQL> SELECT * FROM HR.EMPLOYEES WHERE rownum <2;

EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
EMAIL                     PHONE_NUMBER        HIRE_DATE JOB_ID
     SALARY
```

```
------------------------ -------------------- --------- -------
-
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------
    198 Donald        OConnell
DOCONNEL          650.507.9833          21-JUN-07 SH_CLERK
2600
                124        50

SQL>
```

6. In your `leo_dvowner` session, update the `HR_Application` realm. Change the regular realm type to a MANDATORY realm type `1`. Then review the updated protection.

```
SQL> EXEC DVSYS.DBMS_MACADM.UPDATE_REALM(-
realm_name     => 'HR_Application', -
description    => 'Mandatory Realm to protect the HR
application', -
enabled        => DBMS_MACUTL.G_YES, -
audit_options => DBMS_MACUTL.G_REALM_AUDIT_FAIL +
DBMS_MACUTL.G_REALM_AUDIT_SUCCESS, -
realm_type     => 1)

> > > > >
PL/SQL procedure successfully completed.

SQL> SELECT r.name, o.object_name, r.realm_type
     FROM dvsys.dba_dv_realm r,dvsys.dba_dv_realm_object o
     WHERE r.name = o.realm_name
     AND   o.object_name = 'EMPLOYEES';
  2    3    4
NAME                 OBJECT_NAME           REALM_TYPE
-------------------- -------------------- --------------------
HR_Application       EMPLOYEES             MANDATORY

SQL>
```

7. In another terminal window, test access as the `HR` user to the `EMPLOYEES` table by querying one row.

```
SQL> SELECT * FROM HR.EMPLOYEES WHERE rownum <2;
SELECT * FROM HR.EMPLOYEES WHERE rownum <2
              *
ERROR at line 1:
ORA-01031: insufficient privileges
SQL>
```

Note: The HR user can no longer access the table because he or she has no access to the mandatory HR_Application realm.

8. As the leo_dvowner, describe the DVSYS.DBA_DV_REALM view and then use it to list all realms beginning with HR.

```
SQL> desc DVSYS.DBA_DV_REALM
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 NAME                                      NOT NULL VARCHAR2(128)
 DESCRIPTION                                        VARCHAR2(1024)
 AUDIT_OPTIONS                             NOT NULL NUMBER
 REALM_TYPE                                         VARCHAR2(20)
 ENABLED                                   NOT NULL VARCHAR2(1)

SQL> SELECT name, realm_type, enabled
     FROM DVSYS.DBA_DV_REALM
     WHERE NAME LIKE 'HR%';
  2    3
NAME                 REALM_TYPE                E
-------------------- ------------------- -
HR_Application        MANDATORY          Y
HR_Manager                 REGULAR             Y
HR_Schema            REGULAR            Y

SQL>
```

Note:
In Practice 5-1 you created the regular HR_Schema realm.
Practice 5-2 is optional, so you may or may not have the HR_Manager realm.
In the above Practice 5-3, you created the mandatory HR_Application realm
In the above step 7, you tested access as the HR user to the EMPLOYEES table and received an ORA-01031: insufficient privileges error. This error occurs because the more stringent protection takes precedence.

9. Confirm that the HR user can still select a row from the HR.DEPARTMENTS table which is not protected by a mandatory realm. Then exit the HR SQL*Plus session.

```
SQL> SELECT * FROM HR.DEPARTMENTS WHERE rownum <2;

DEPARTMENT_ID DEPARTMENT_NAME              MANAGER_ID LOCATION_ID
------------- ------------------------------ ---------- ---------
           10 Administration                 200        1700

SQL> exit
$
```

10. As the `leo_dvowner` user, delete the HR_Application realm and the HR_Manager realm, but keep the regular HR_Schema realm, which will be used in the following practice. Then exit all terminal windows.

```
SQL> show user
USER is "LEO_DVOWNER"
SQL> EXEC DVSYS.DBMS_MACADM.DELETE_REALM(realm_name =>
'HR_Application')

PL/SQL procedure successfully completed.

SQL> EXEC DVSYS.DBMS_MACADM.DELETE_REALM(realm_name =>
'HR_Manager')

PL/SQL procedure successfully completed.

SQL> exit
$ exit
```

Practices for Lesson 6: Defining Rule Sets

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

In these practices, you will create rule sets. You then associate the rule sets with a realm to ensure access only under certain circumstances.

Assumptions

Practices 3-1, 3-2 and 3-4 were successfully completed:

- Database Vault is configured.
- Test users are created.
- The `leo_dvowner` user is configured to have access to Database Vault via Cloud Control.

Practice 6-1: Managing Rule Sets

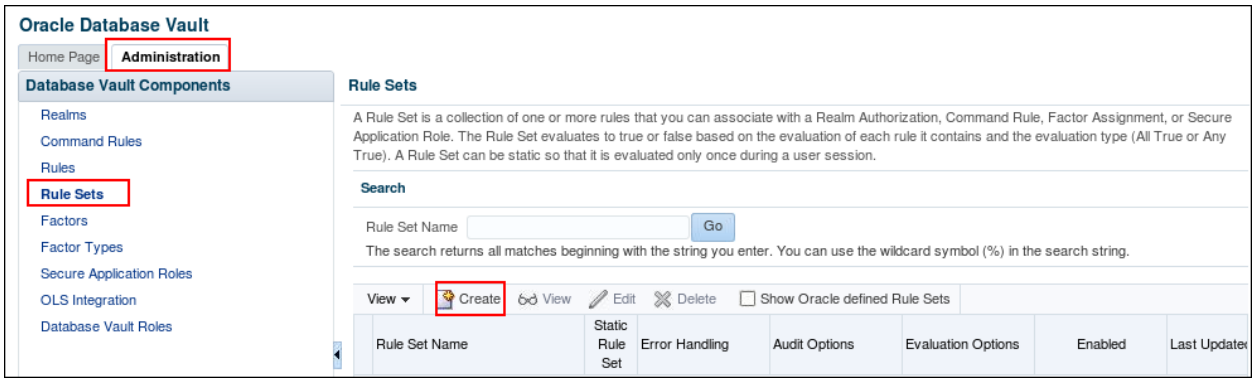
Overview

In this practice, you assume that you have a task that must be done OUTSIDE of the regular work hours. Create a rule set called `Non_Work_Hours` that enforces the rule that access is allowed only during nonworking hours of a day, which include any hours outside of 8:00 AM and 4:59 PM local time, Monday through Friday. Assume that the server is in the local time zone. Use two rules: `Night` and `Weekend`.

Tasks

1. As `leo_dvowner`, use Enterprise Manager Cloud Control to create the `Non_Work_Hours` rule set. Set the Evaluation Options to **Any True**, because you create multiple rules for this rule set and if any of these rules evaluate to true, the rule set evaluates to true. The first rule is called `Night` and has the expression `to_char(sysdate,'hh24') not between '08' and '16'`.
   a. Navigate to Oracle Database Vault Administration > Rule Sets > Create (button).
   b. Enter `Non_Work_Hours` as **Rule Set Name**, (optionally) enter a description, accept **Enabled**, and set the **Evaluation Options** to **Any True**. Then click **Next**.
   c. On the "Create Rule Set: Associate with Rules" page, click **Create Rule**.
   d. Enter `Night` as **Name** and `to_char (sysdate,'hh24') not between '08' and '16'` as **Expression**. Then click **OK**.
   e. Confirm the rule name and expression and click **Next**.
   f. Accept the default error-handling and audit options and click **Next**.
   g. Review the `Non_Work_Hours` rule set, especially the code at the bottom of the page. Then click **Finish**.
   h. You should receive a success message.

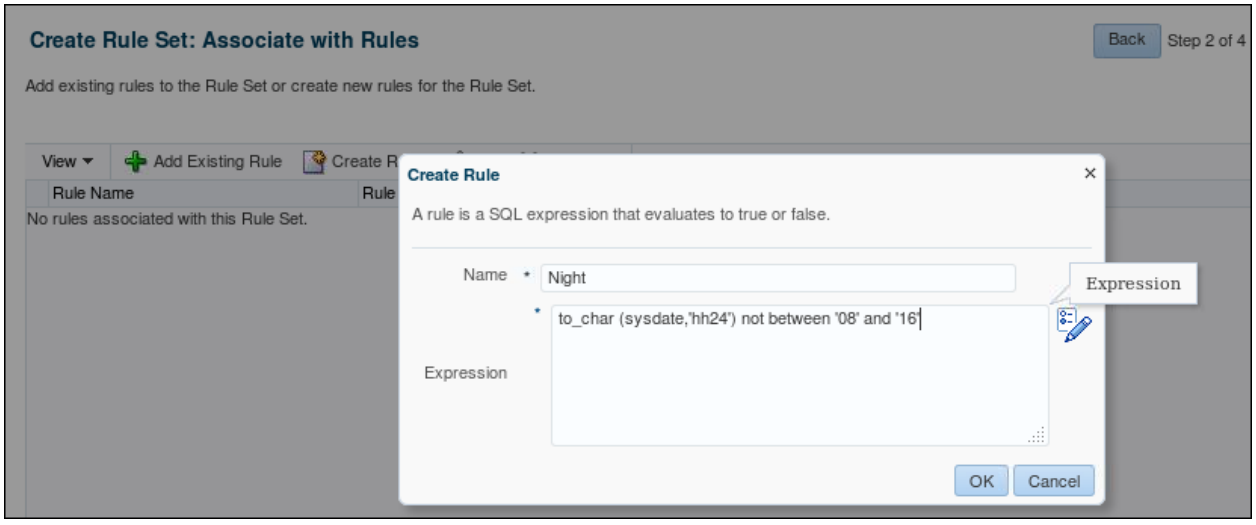
a.



b.

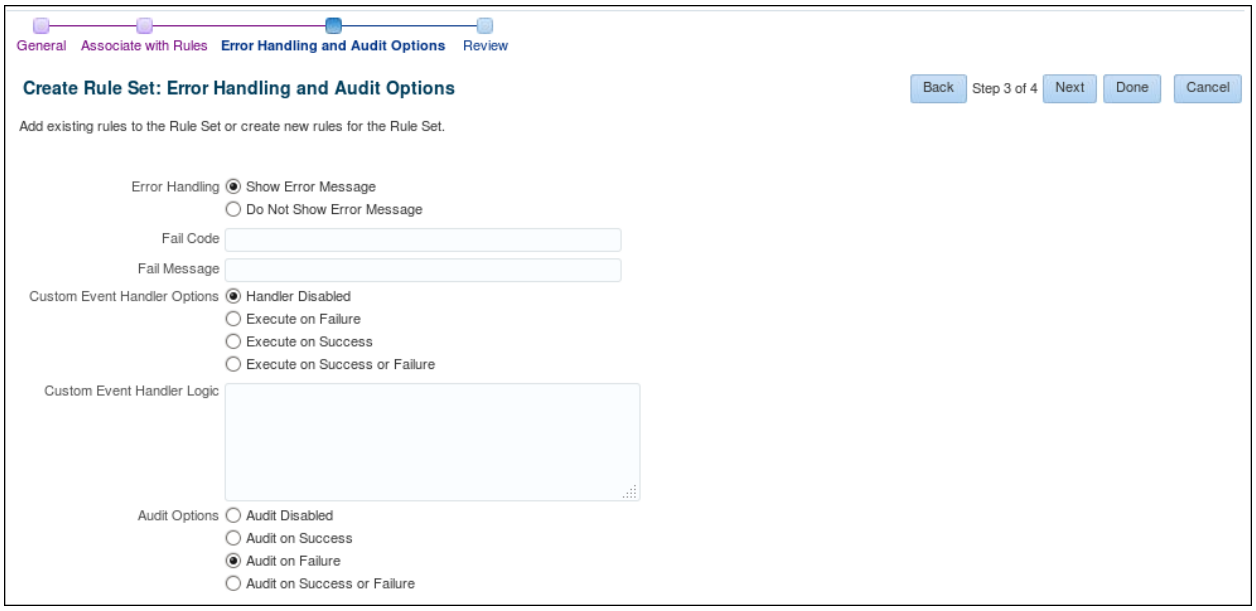
c.

d.



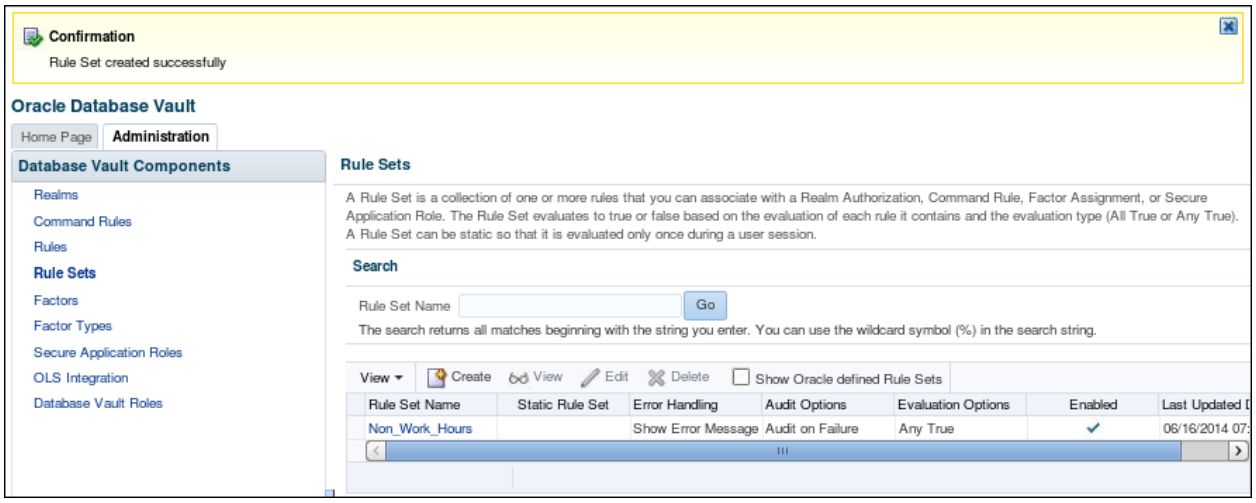
e.

f.



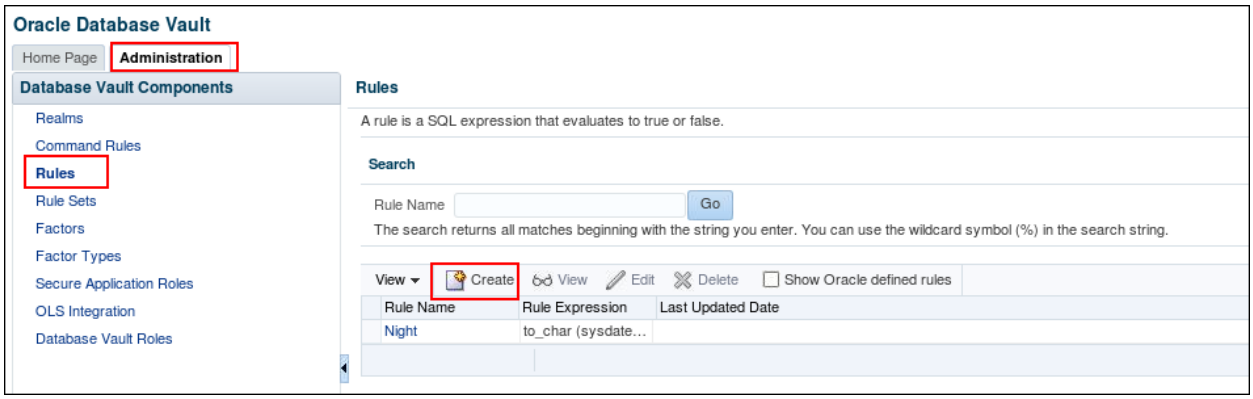
g.

h.

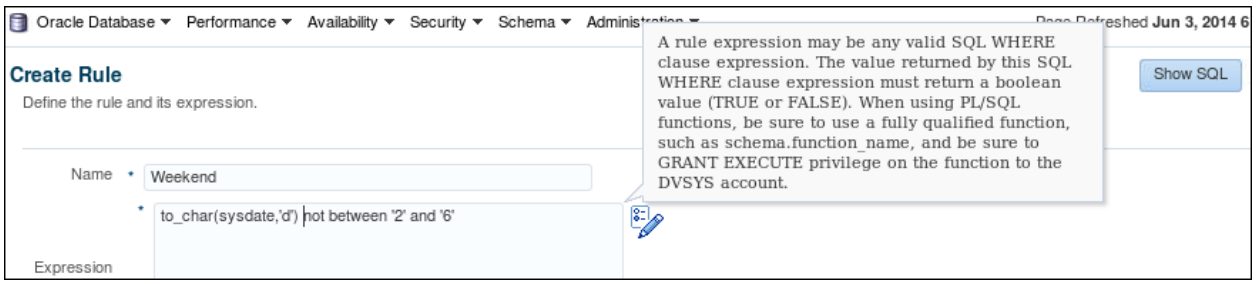


2. Still as `leo_dvowner`, use Enterprise Manager Cloud Control to create the second `Weekend` rule with the expression `to_char(sysdate,'d') not between '2' and '6'`.
   a. Navigate to Oracle Database Vault Administration > Rules > Create (button).
   b. Enter `Weekend` as **Name** and `to_char(sysdate,'d') not between '2' and '6'` as **Expression**.
   c. Optionally, click **Show SQL**, review the code, and then click **OK**.
   d. Click **OK** to create the rule. You should receive a success message.

a.

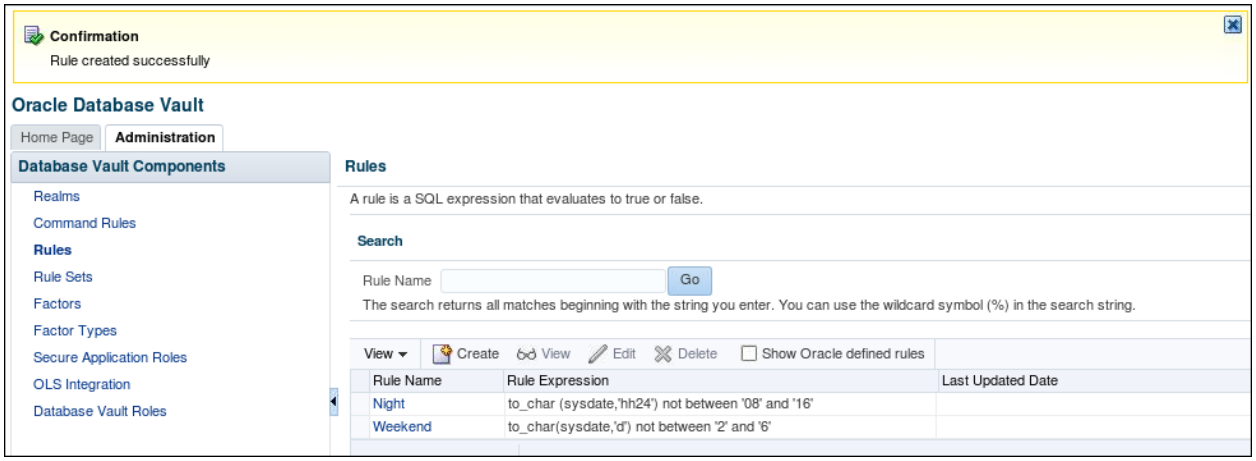


b.



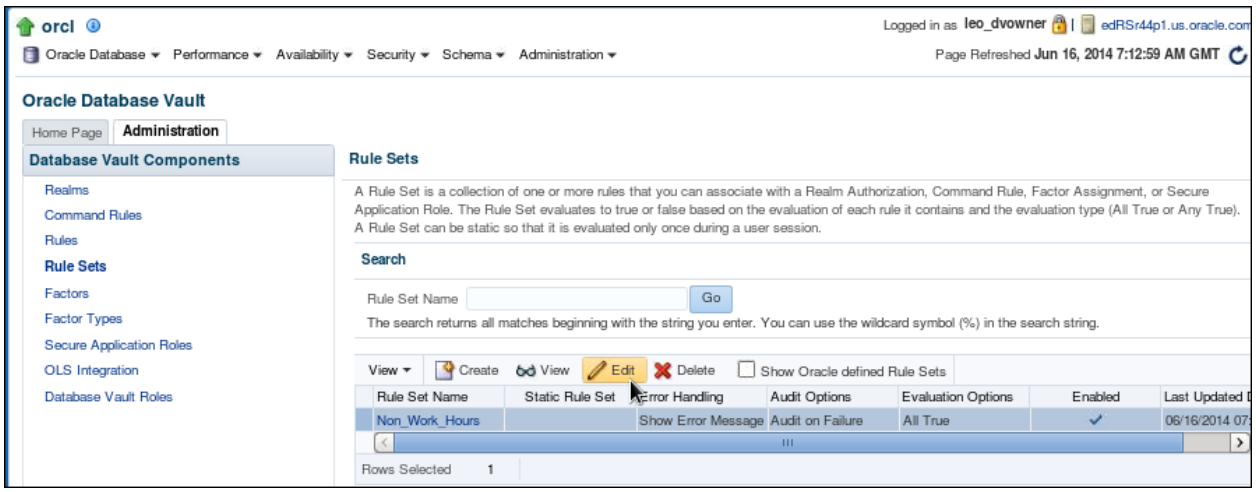
c.

d.

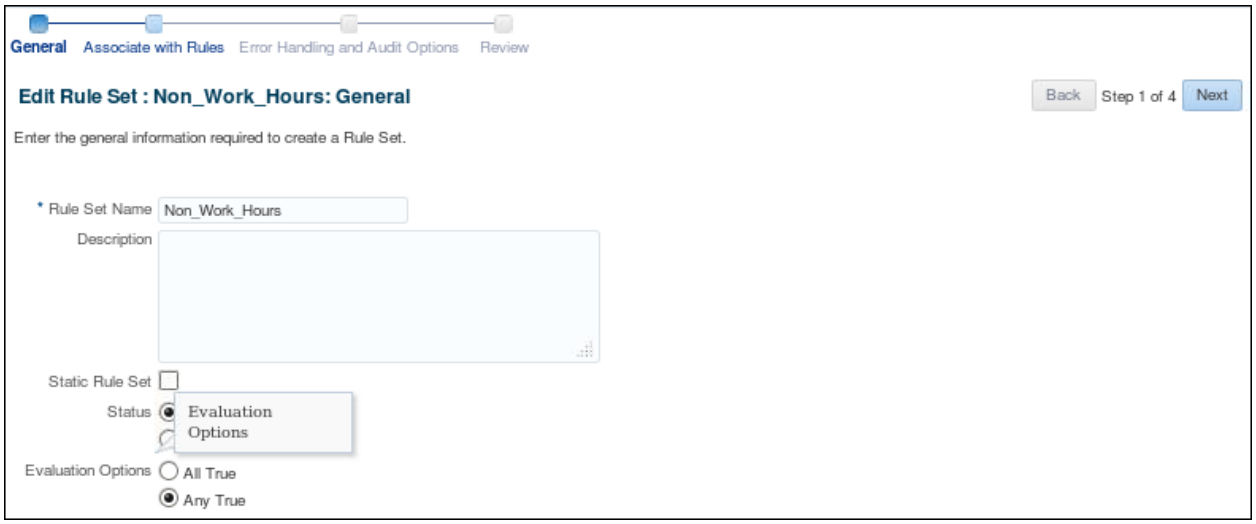


3. As the `leo_dvowner` user, attach the `Weekend` rule to the `Non_Work_Hours` rule set.
   a. Navigate to Oracle Database Vault Administration > Rule Sets. Select `Non_Work_Hours` and click the **Edit** button.
   b. On the Edit Rule Set: Non Work Hours: General page click **Next**.
   c. On the "Associate with Rules" page, click **Add Existing Rules**. Select the `Weekend` rule and click **OK**.
   d. Confirm that you now have two rules associated with your rule set, and click **Next**.
   e. Accept the default error-handling and audit options, and click **Next**.
   f. Review the `Non_Work_Hours` rule set, especially the code at the bottom of the page. Then click **Finish**.
   g. You should receive a success message.

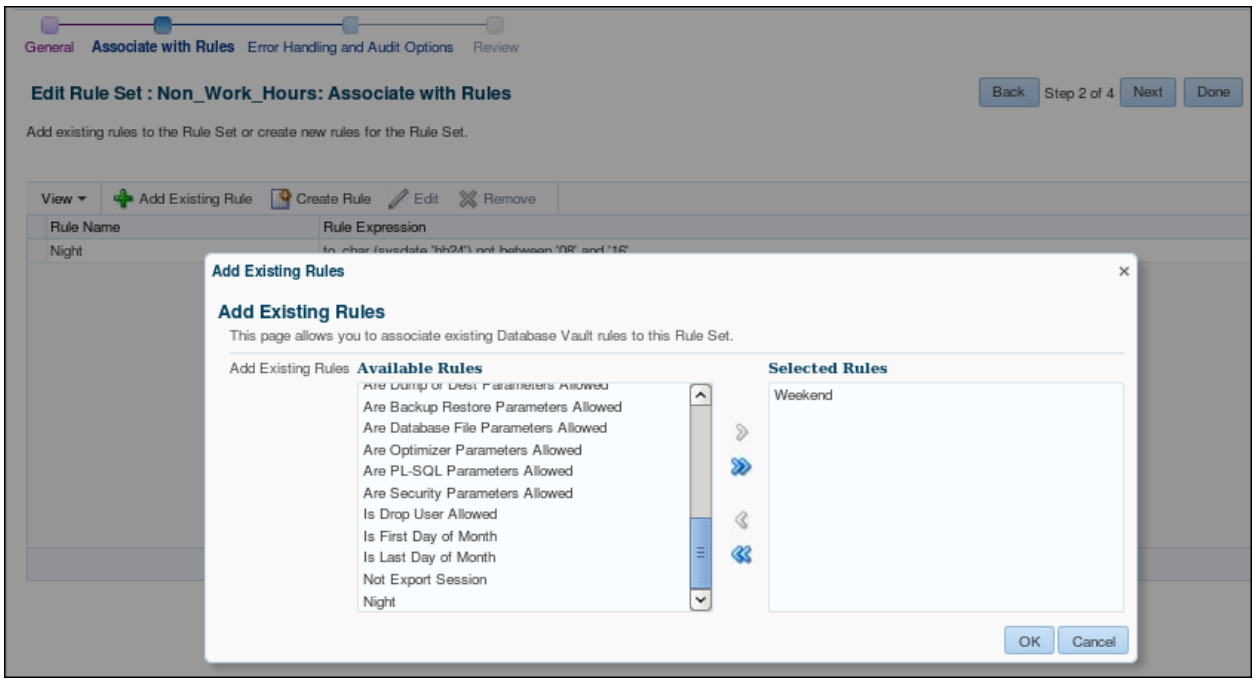
a.



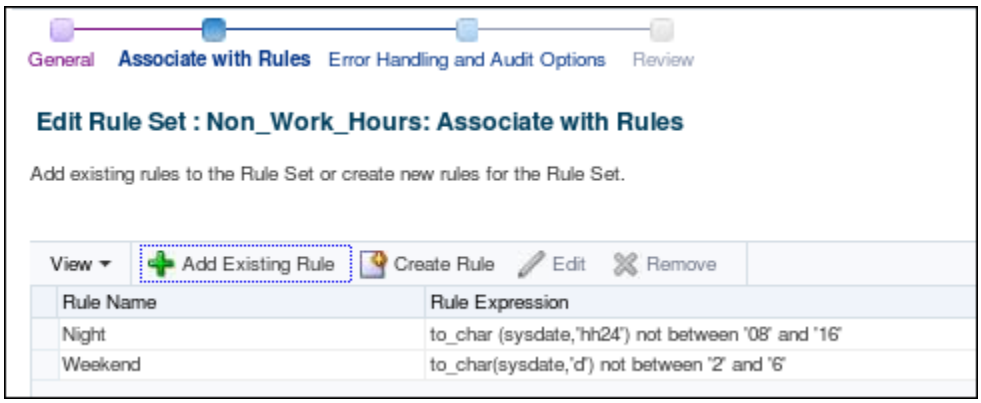
b.



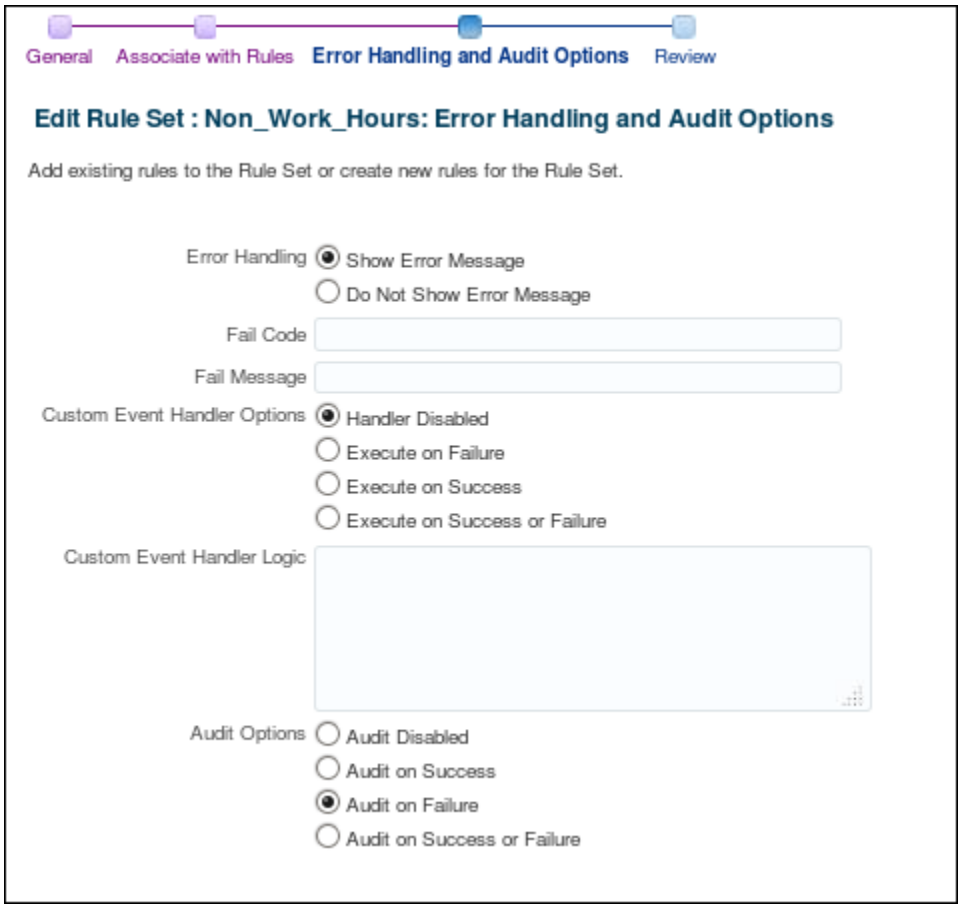
c.



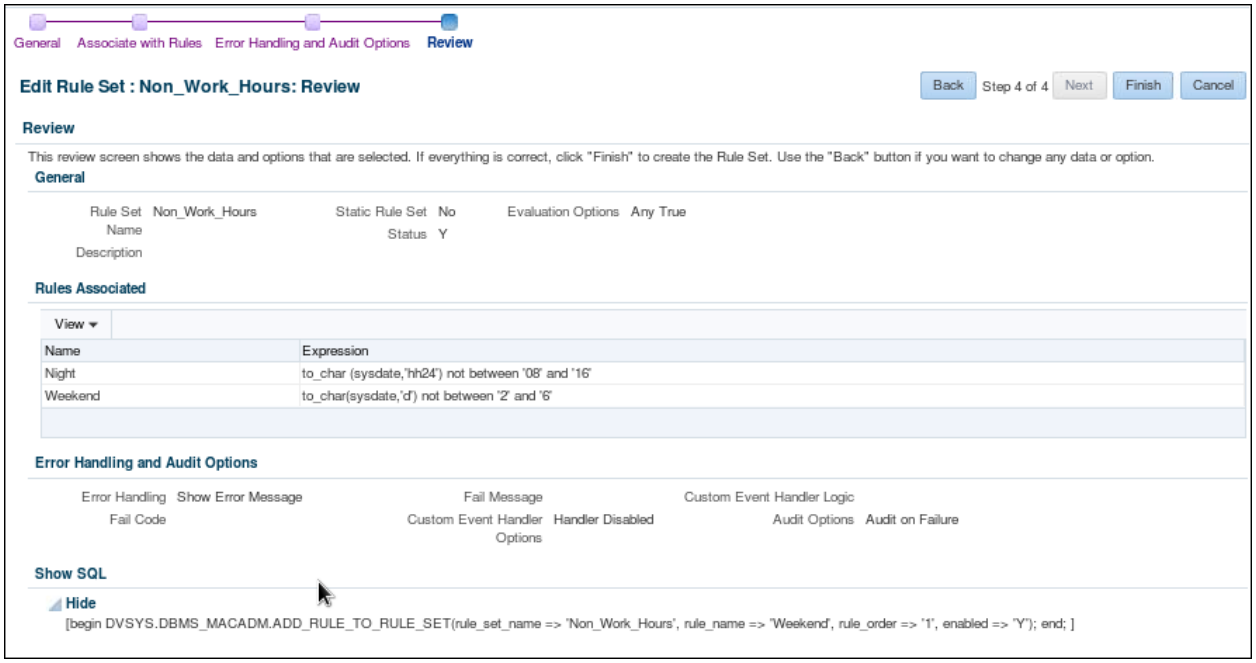
d.



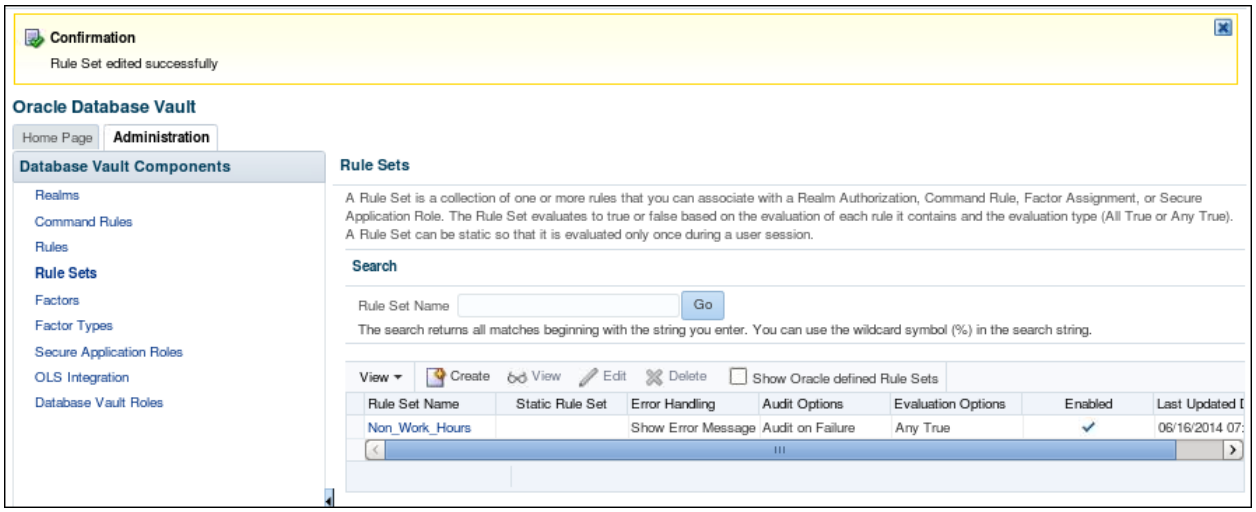
e.



f.

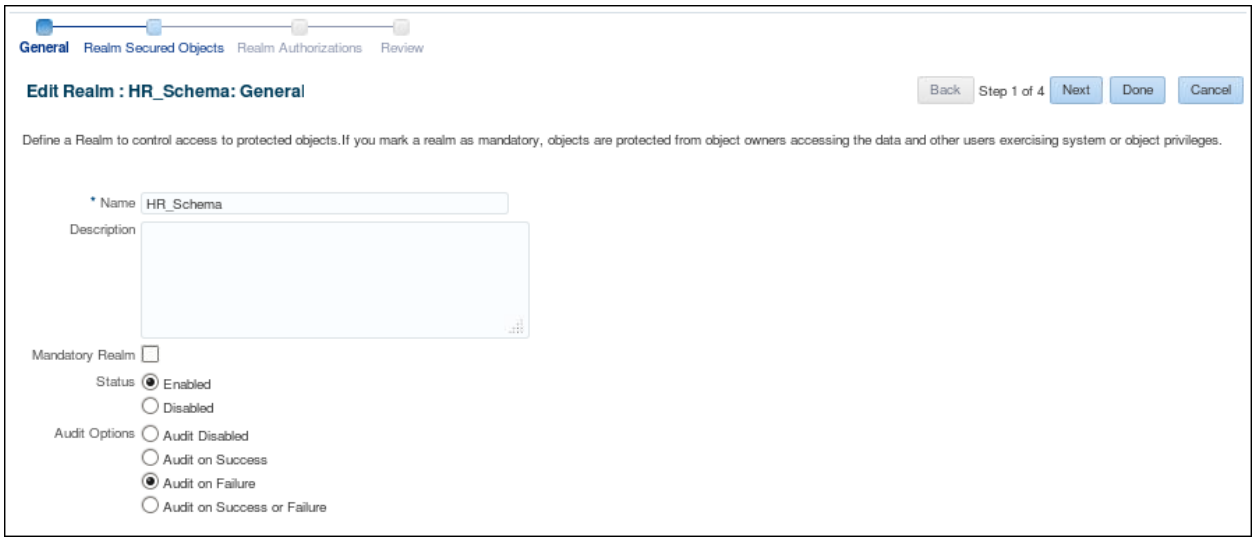


g.

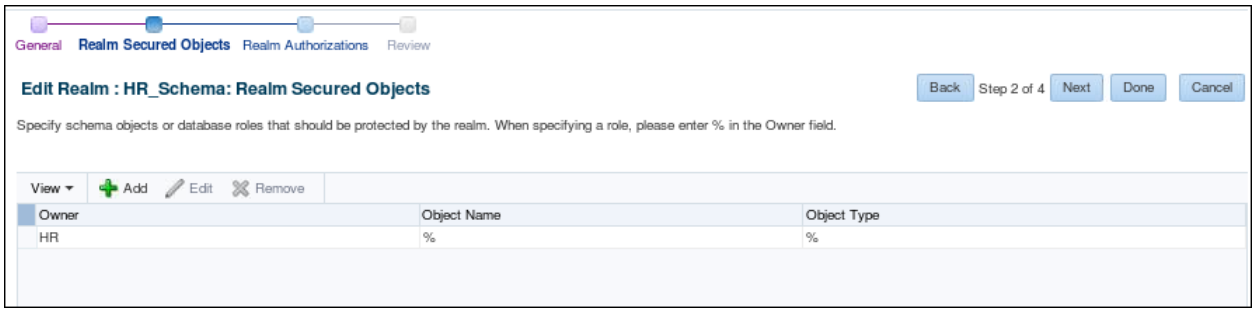


4. As the `leo_dvowner` user, use the `Non_Work_Hours` rule set to restrict DBA activity on the `HR_Schema` to only those hours defined by the `Non_Work_Hours` rule set. To do this, edit the `HR_Schema` realm and add the `Non_Work_Hours` rule set as an authorization rule set for the `BERNST` participant.
   a. Navigate to Oracle Database Vault Administration > Realms. Select `HR_Schema` and click the **Edit** button. On the Edit Realm: HR_Schema: General page, click **Next**.
   b. On the Edit Realm: HR_Schema: Realm Secured Objects page, click **Next**.
   c. On the Edit Realm: HR_Schema: Realm Authorizations page, select `BERNST` and click the **Edit** button. In the Edit Authorization window for `BERNST`, select or enter `Non_Work_Hours` as **Realm Authorization Rule Set**, and click **OK**.
   d. On the Edit Realm: HR_Schema: Realm Authorizations page, click **Next**.
   e. Review the `HR_Schema` realm, especially the code at the bottom of the page. Then click **Finish**.
   f. You should receive a success message.

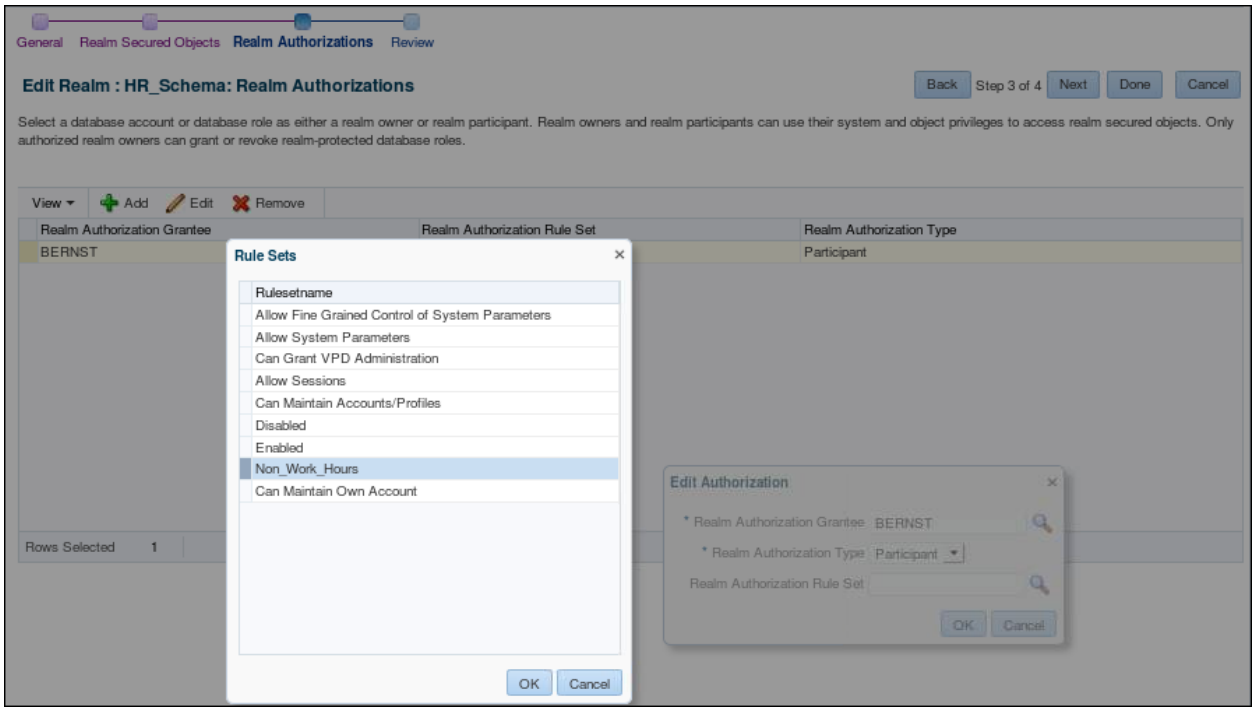
a.



b.

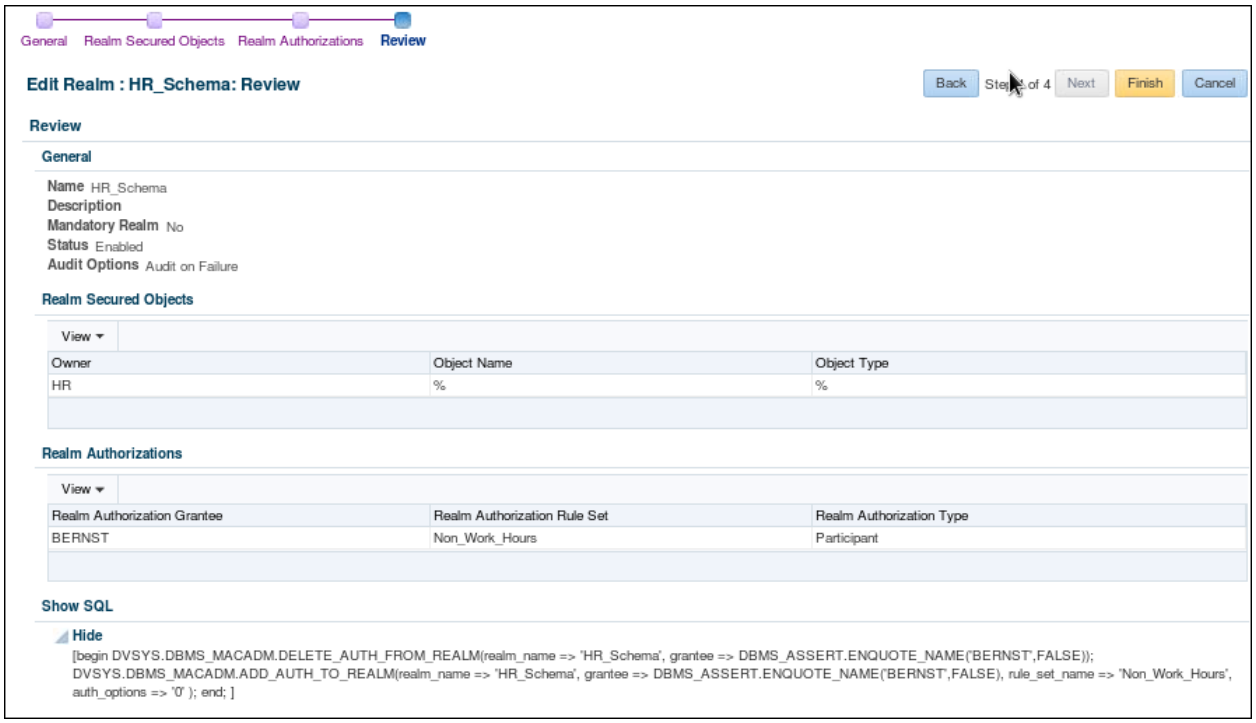


c.

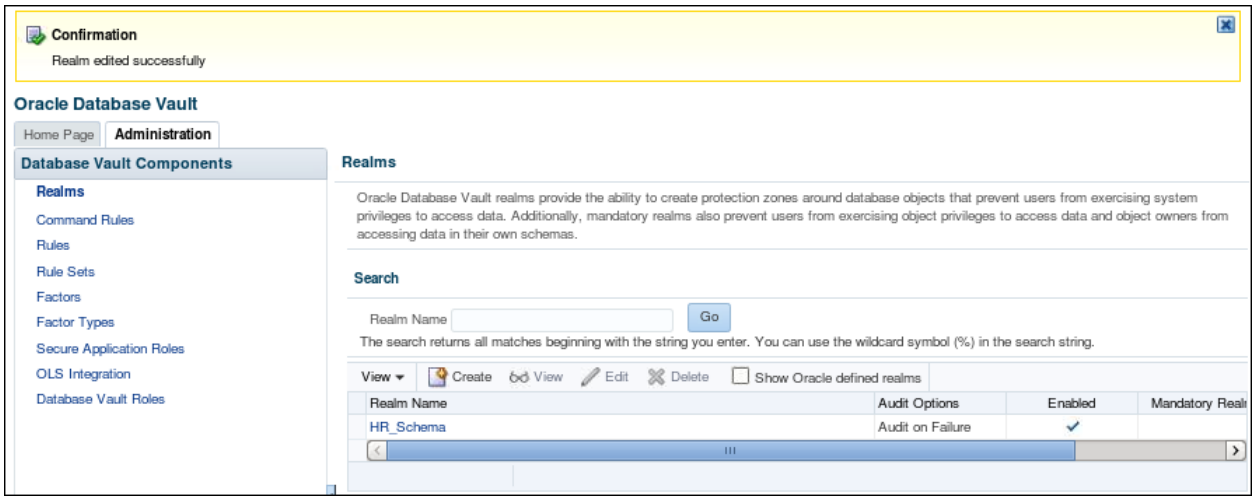


d.

e.



f.



5. To test the results of your restriction, log in to SQL*Plus as the **`BERNST`** user, check your system date with **`!date`**, and attempt to create a new test table. Then exit.

Question: What happens and why?

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus bernst
Enter password:
SQL> !date
Mon Jun 16 07:22:16 UTC 2014
SQL> create table hr.x (a number);

Table created.

SQL> drop table hr.x;

Table dropped.

SQL>
```

Answer: The table is created because the command is tested outside the regular work hours. Drop the table if you can create it.
Answer: If you are inside the work hours, then you get the `ORA-47401: Realm violation` error message.

Note: Retest at a different point in time to see both behaviors.

```
SQL> create table hr.x (a number);
create table hr.x (a number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on HR.X

SQL> drop table hr.x;
drop table hr.x
*
ERROR at line 1:
ORA-47401: Realm violation for DROP TABLE on HR.X

SQL> exit
$
```

Important: Reset the rules expressions to the initial values defined in tasks 1 and 2, if you changed them for testing purposes.

Practices for Lesson 7: Configuring Command Rules

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

In these practices, you will work with command rules.

- First, you use a command rule to prevent the creation of a view.
- Then you create a rule set that prevents a user from selecting his or her own data, but still allows the user to alter and create objects.

This is the method to prevent an application administrator from reading the data in the objects that he or she manages.

Finally, you create a command rule designed to prevent the accidental execution of a command. This is done by forcing the command to be capitalized in a special way.

Assumptions

Practices 3-1, 3-2, and 3-4 were successfully completed:

- Database Vault is configured.
- Test users are created.
- The `leo_dvowner` user is configured to have access to Database Vault via Cloud Control.

Practice 7-1: Using Command Rules

Overview

In this practice, you create a command rule to prevent the creation of a view.

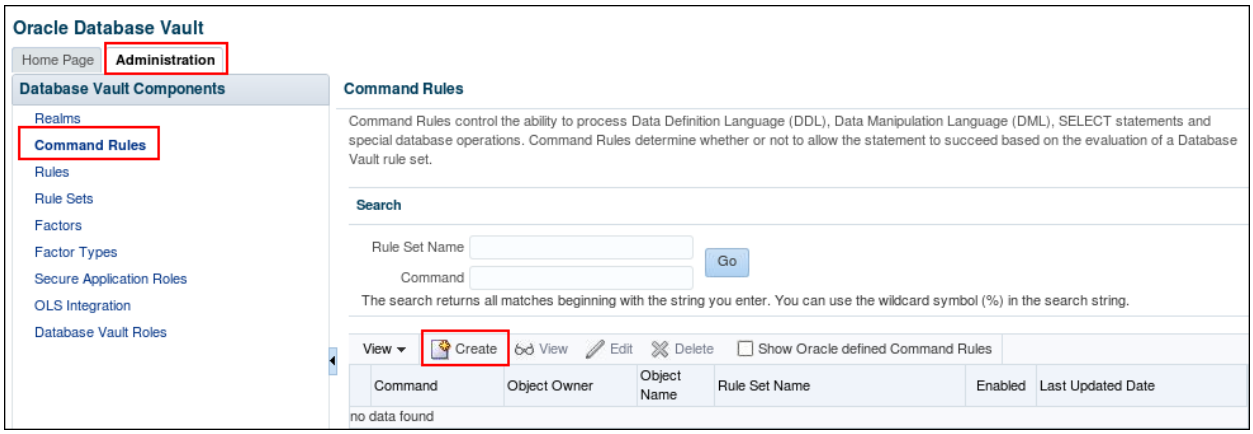
Tasks

1. Before you create the command rule, confirm that AHUNOLD can create a view. Create the OE.BIG_ORDERS view that is defined as all OE.ORDERS rows where ORDER_TOTAL is greater than 100000. Then drop the view so that you can attempt to create it again later.

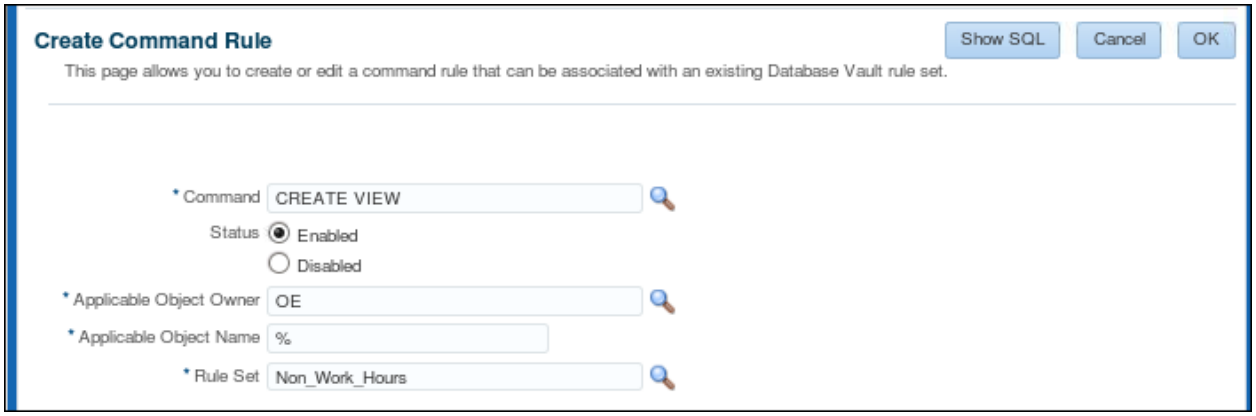
```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus ahunold
Enter password:
SQL>
SQL> CREATE VIEW oe.big_orders AS SELECT * FROM oe.orders
     WHERE       order_total > 100000;
  2
View created.

SQL> DROP VIEW oe.big_orders;

View dropped.

SQL>
```

2. As the leo_dvowner user, use Enterprise Manager Cloud Control to create a command rule that prevents users from issuing the CREATE VIEW command on OE objects unless it is during nonworking hours. Use the Non_Work_Hours rule set that you created earlier.
   a. Navigate to Oracle Database Vault Administration > Command Rules > Create (button).
   b. Enter or select CREATE VIEW as **Command**, accept **Enabled**, enter or select OE as **Applicable Object Owner**, accept % as **Applicable Object Name**, and enter or select Non_Work_Hours for **Rule Set**.
   c. Optionally, click **Show SQL**, review the code, and then click **OK**.
   d. On the Create Command Rule page, click **OK**. - You should receive a success message.

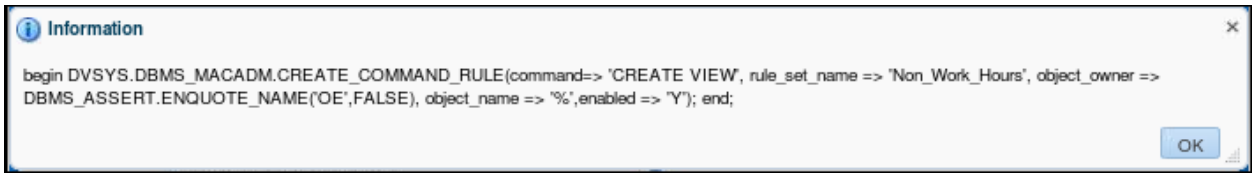
a.



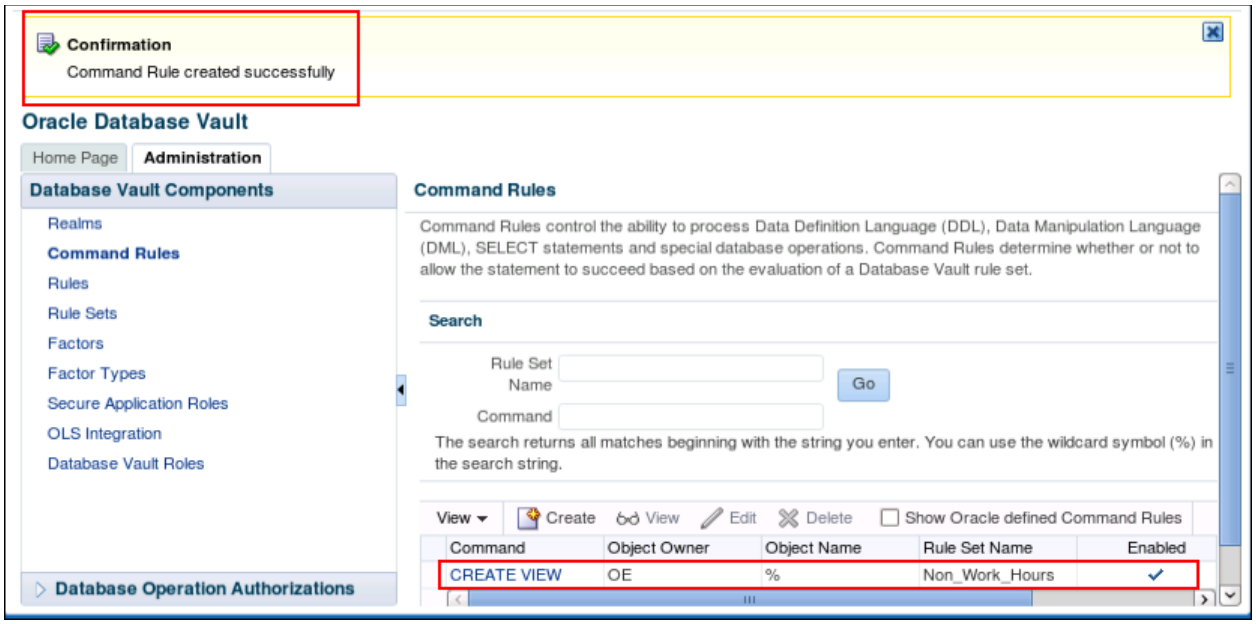
b.



c.



d.



3. In the SQL*Plus session as the **AHUNOLD** user try issuing the **CREATE VIEW** command.

   a. *Question:* What happens and, why?

```
SQL> CREATE VIEW oe.big_orders AS SELECT * FROM oe.orders
     WHERE  order_total > 100000;
  2
CREATE VIEW oe.big_orders AS SELECT * FROM oe.orders
                                          *
ERROR at line 1:
ORA-47400: Command Rule violation for CREATE VIEW on
OE.BIG_ORDERS


SQL>
```

   Answer: The `CREATE VIEW` command fails when issued within the work hours, it succeeds only outside the work hours, because the command rule just created requires that `CREATE VIEW` be issued only during nonworking hours.

   b. If you wish to confirm your system time, enter `!date`.

```
SQL> !date
Wed Jul  9 20:23:29 UTC 2014

SQL> CREATE VIEW oe.big_orders AS SELECT * FROM oe.orders
     WHERE  order_total > 100000;
```

```
  2
View created.
SQL>
```

c. Drop the view, if you create it.

```
SQL> DROP VIEW oe.big_orders;

View dropped.
SQL>
```

Practice 7-2: Protecting Application Data

Overview

In this optional practice, you prevent an application DBA from reading the data in the objects that he or she manages. To accomplish this:

- You create a database role called `APP_ROLE`.
- You then create a rule set containing a rule to ensure that the user has the `APP_ROLE` role.
- Finally, you create a command rule to ensure that the `SELECT` command for any object in the `OE` schema succeeds only if the user has the `APP_ROLE` role.

Tasks

1. Begin by showing a potential security issue. The `AHUNOLD` user can view data from the schema he or she manages.

```
SQL> show user
USER is "AHUNOLD"
SQL> select order_id, customer_id, order_total
     from   oe.orders
     where  rownum < 8;
  2    3
  ORDER_ID CUSTOMER_ID ORDER_TOTAL
---------- ----------- -----------
      2458         101     78279.6
      2397         102     42283.2
      2454         103      6653.4
      2354         104       46257
      2358         105        7826
      2381         106     23034.6
      2440         107     70576.9

7 rows selected.

SQL>
```

Note: Assume your current security requirements forbid the display of application data to the DBA.

2. As the `AHUNOLD` user, create a new test table. Drop the table. These commands should succeed.

```
SQL> create table oe.test (customer_id NUMBER(12));

Table created.

SQL> drop table oe.test;
```

```
Table dropped.

SQL>
```

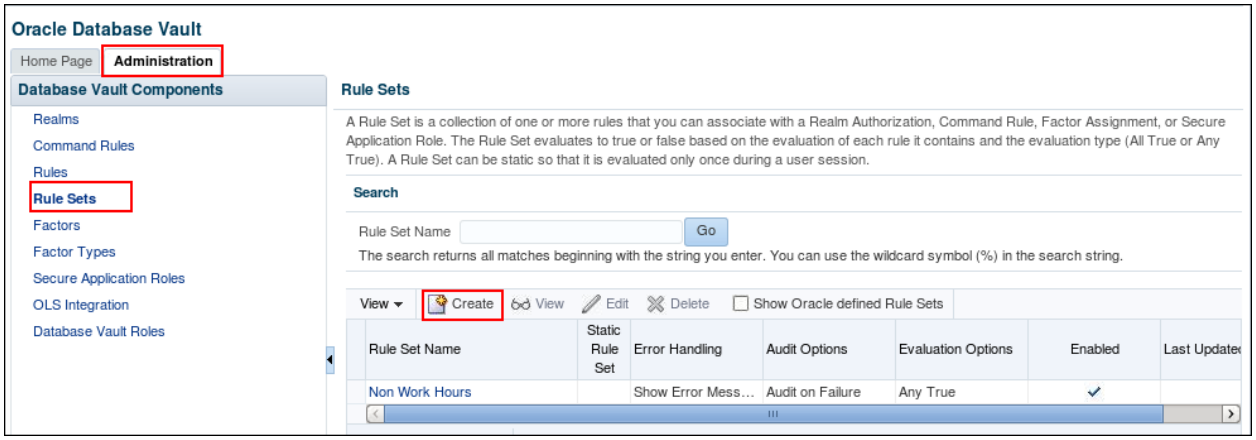
Note: Your current requirements allow the creation of objects only outside of the regular business hours.

3. Connect to SQL*Plus as `DBA_PSMITH` and create a (traditional) role called `APP_ROLE`.

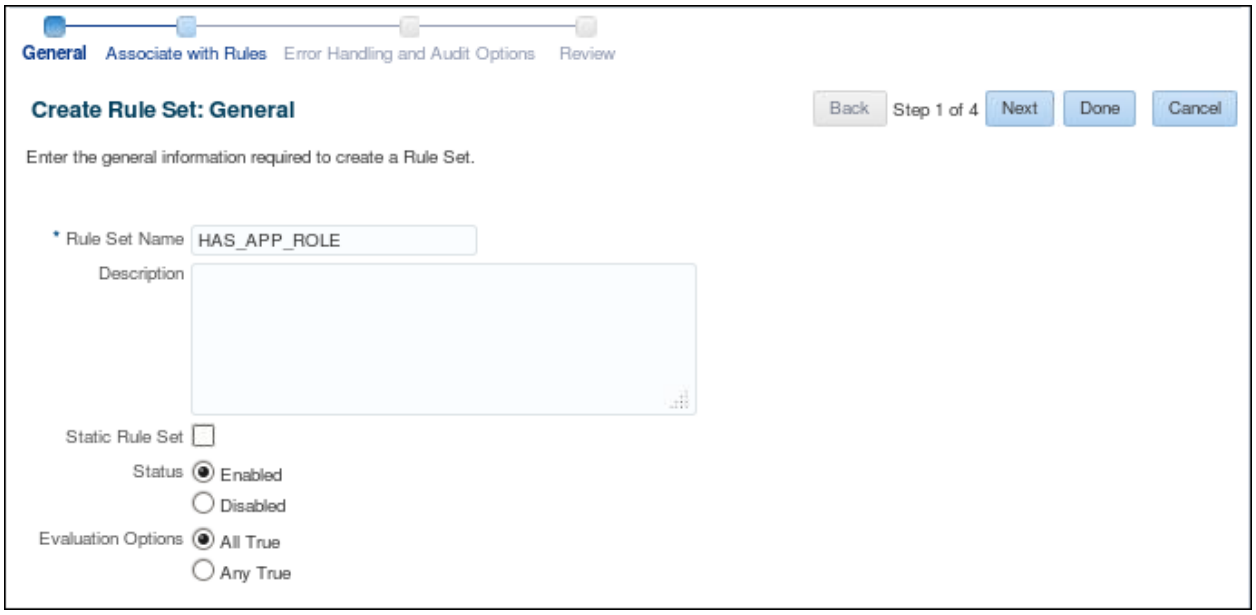
```
SQL> connect dba_psmith
Enter password:
Connected.
SQL> CREATE ROLE app_role;

Role created.
SQL>
```

4. As the `leo_dvowner` user, use Enterprise Manager Cloud Control to create the `HAS_APP_ROLE` rule set. Set the Evaluation Option to **All True**. The rule set contains a rule with the following expression:
`DVSYS.DBMS_MACUTL.USER_HAS_ROLE_VARCHAR('APP_ROLE') = 'Y'`
   a. Navigate to Oracle Database Vault Administration > Rule Sets > Create (button).
   b. Enter `HAS_APP_ROLE` as **Rule Set Name**, an optional description, accept **Enabled**, and set the **Evaluation Options** to **All True**. Then click **Next**.
   c. On the "Create Rule Set: Associate with Rules" page, click **Create Rule**.
   d. Enter `Has_App_User_Role` as **Name** and `DVSYS.DBMS_MACUTL.USER_HAS_ROLE_VARCHAR('APP_ROLE') = 'Y'` as **Expression**. Then click **OK**.
   e. Confirm the rule name and expression, and click **Next**.
   f. Accept the default error handling and audit options, and click **Next**.
   g. Review the rule set, especially the code at the bottom of the page, and then click **Finish**.
   h. You should receive a success message.

a.

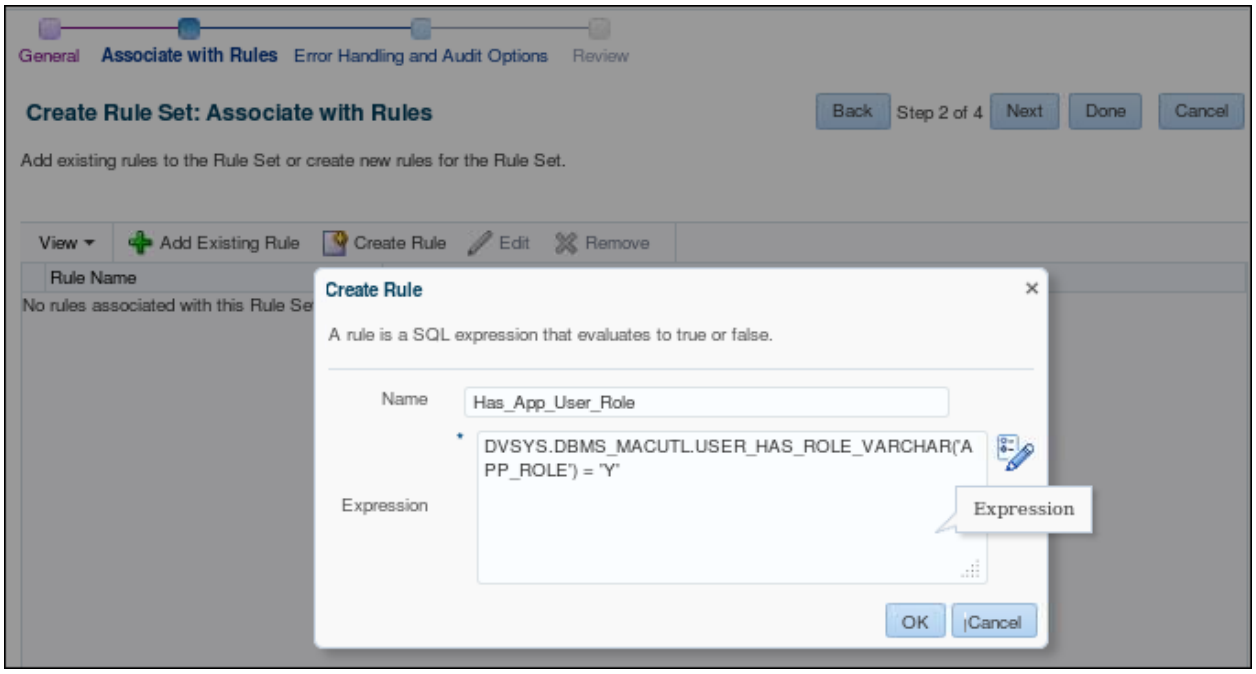


b.



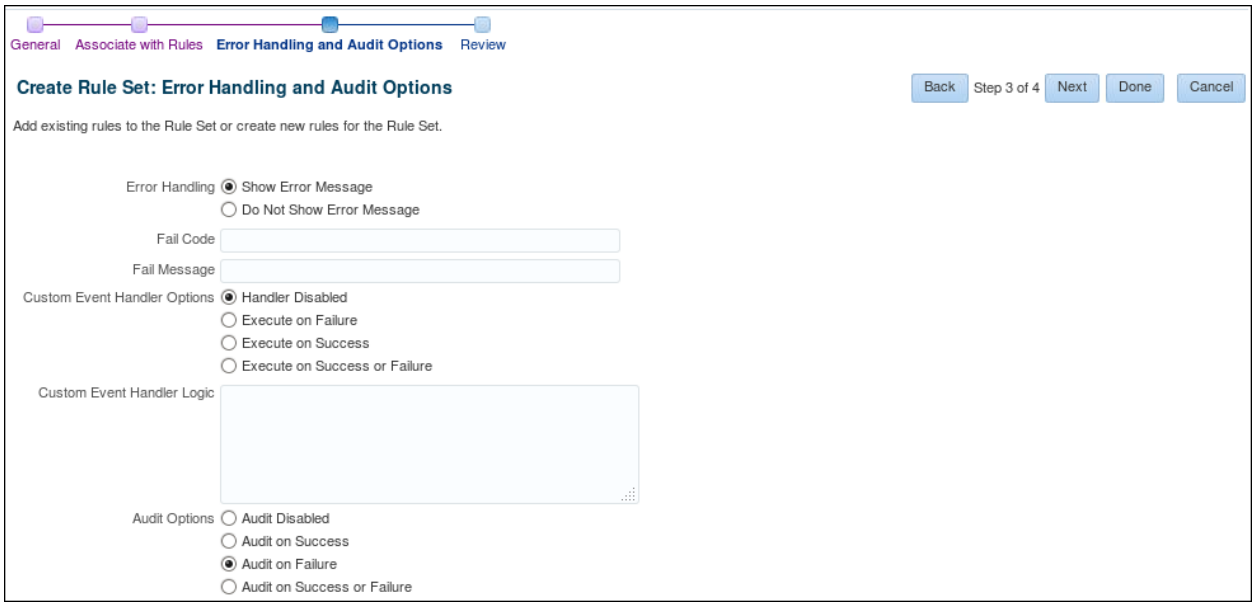
c.

d.

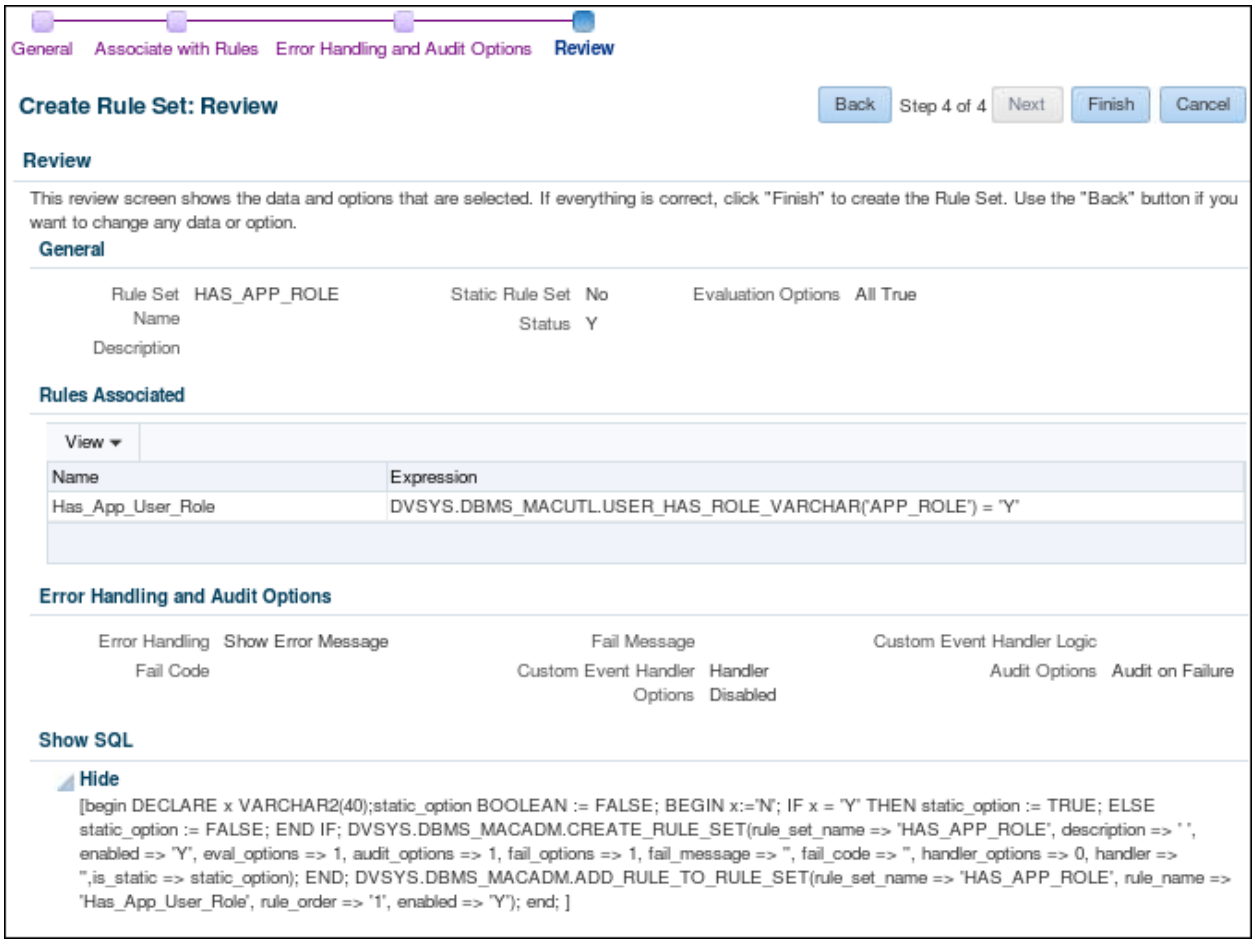


e.

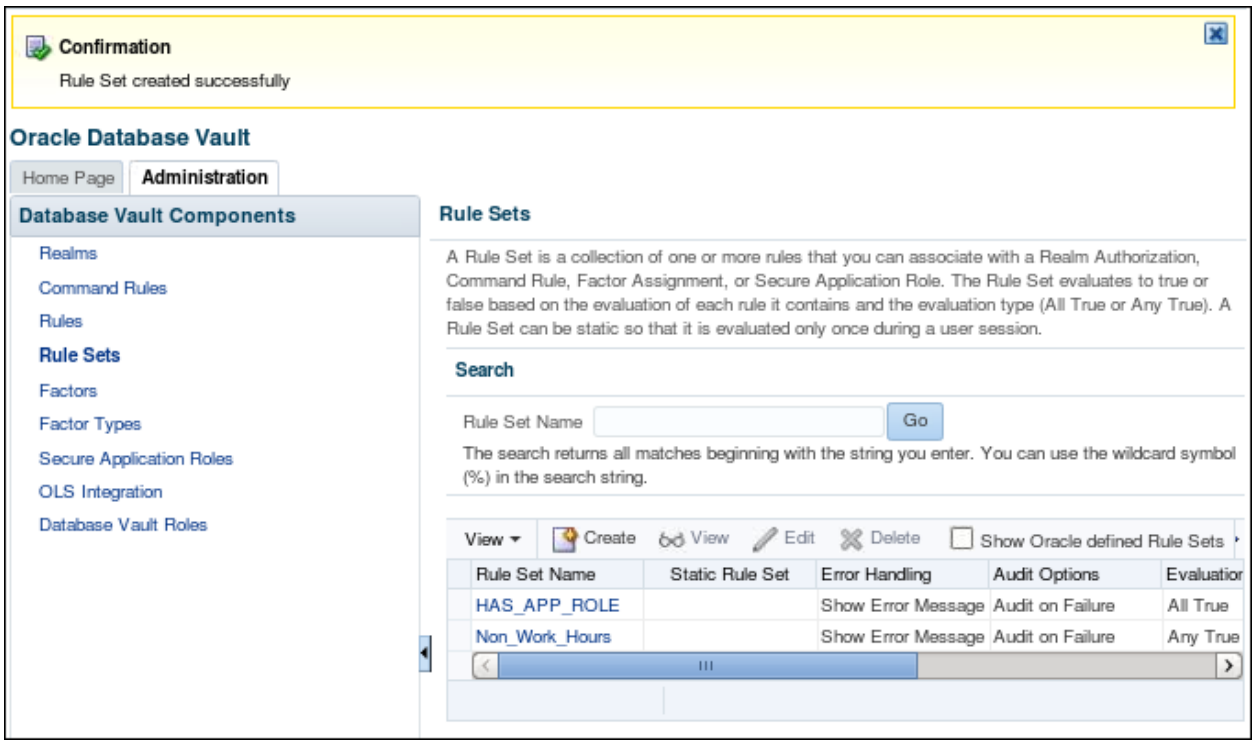
f.



g.

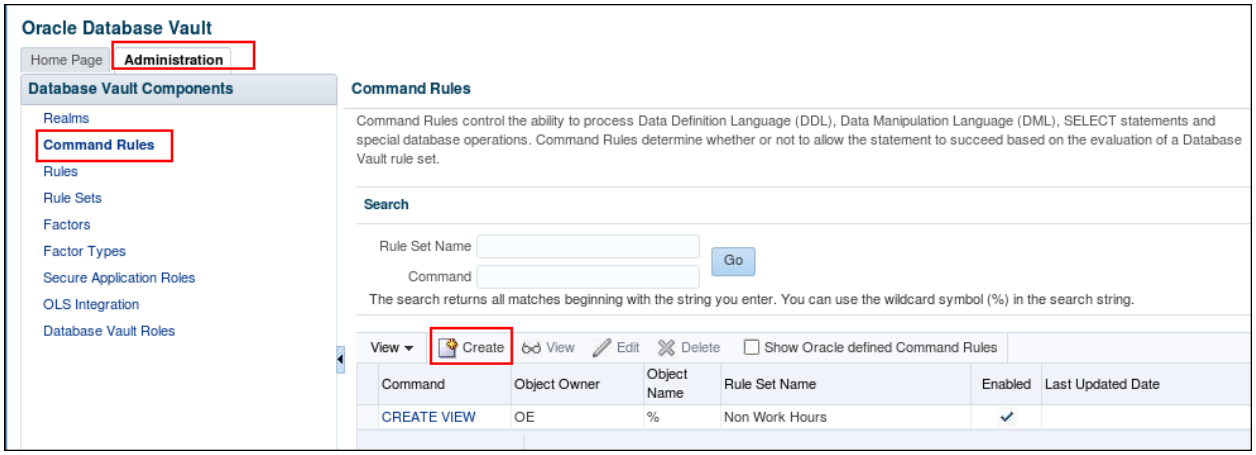


h.

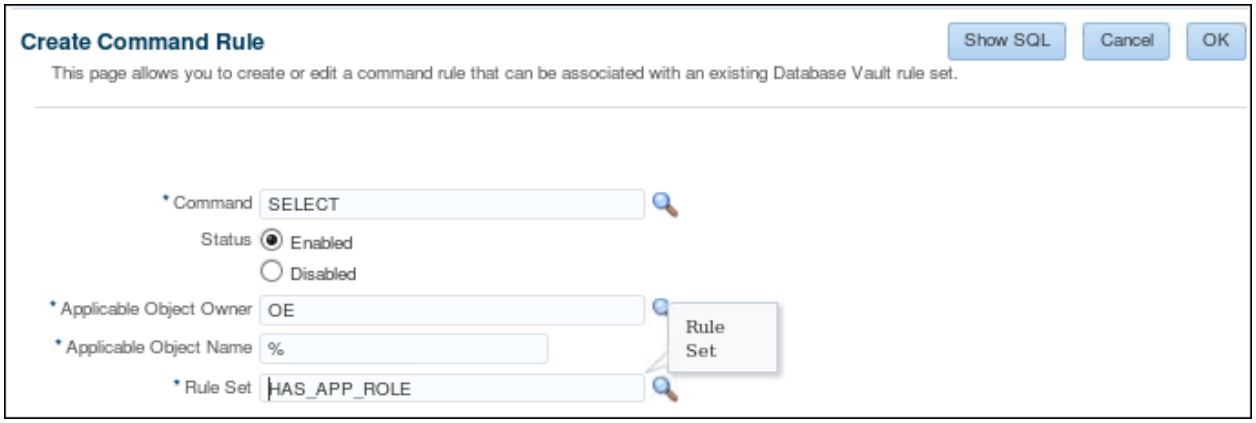


5. As the `leo_dvowner` user, create a command rule for the `SELECT` command. This command rule should apply to all objects in the `OE` schema. Assign the `HAS_APP_ROLE` rule set to this command rule.
   a. Navigate to Oracle Database Vault Administration > Command Rules > Create (button).
   b. Enter or select `SELECT` as **Command**, accept **Enabled**, enter or select `OE` as **Applicable Object Owner**, accept `%` as **Applicable Object Name**, and enter or select `HAS_APP_ROLE` as **Rule Set**.
   c. Optionally, click **Show SQL**, review the code, and then click **OK**.
   d. On the Create Command Rule page, click **OK**.
   e. You should receive a success message and see the `SELECT` command rule.

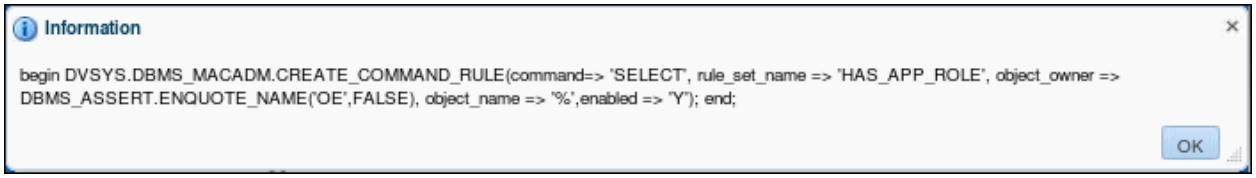
a.



b.

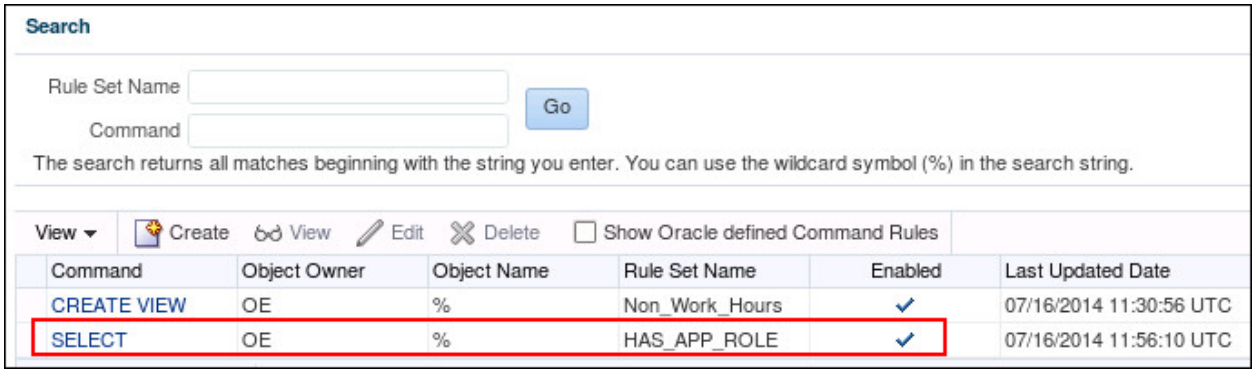


c.



d.

e.



6. Attempt your test tasks again. As the `AHUNOLD` user, query the `OE.ORDERS` table.
   Question: This should fail. Why?

```
SQL> connect ahunold
Enter password:
Connected.
SQL> select order_id, customer_id, order_total
     from   oe.orders
     where  rownum < 8;
  2     3
   from oe.orders
        *
ERROR at line 2:
ORA-01031: insufficient privileges

SQL>
```

   Answer: The command rule causes the `SELECT` command to fail because `AHUNOLD` does not have the `APP_ROLE` role.

7. Confirm that **`AHUNOLD`** can still create objects. As the **`AHUNOLD`** user, create a new test table and drop it. This should still succeed.

```
SQL> create table oe.test (customer_id NUMBER(12)) ;

Table created.
```

```
SQL> drop table oe.test;

Table dropped.

SQL>
```

8. Confirm that it is the absence of the APP_ROLE role that prevents the SELECT access. Do this by granting this role to AHUNOLD as DBA_PSMITH.

```
SQL> connect dba_psmith
Enter password:
Connected.
SQL>
SQL> GRANT app_role to ahunold;

Grant succeeded.
SQL>
```

9. Try the SELECT statement again as AHUNOLD.

```
SQL> connect ahunold
Enter password:
Connected.
SQL> select order_id, customer_id, order_total
     from   oe.orders
     where  rownum < 8;
  2    3
  ORDER_ID CUSTOMER_ID ORDER_TOTAL
---------- ----------- -----------
      2458         101     78279.6
      2397         102     42283.2
      2454         103      6653.4
      2354         104       46257
      2358         105        7826
      2381         106     23034.6
      2440         107     70576.9

7 rows selected.

SQL>
```

10. Then revoke the role as DBA_PSMITH.

```
SQL> connect dba_psmith
Enter password:
Connected.
SQL>
SQL> REVOKE app_role FROM ahunold;

Revoke succeeded.

SQL> EXIT
$
```

Practices for Lesson 8: Extending Rule Sets

Chapter 8

Practices for Lesson 8: Overview

Practices Overview

In these practices, you will work with factors and identities. You will create some factors whose values are evaluated in different ways—one value is computed by looking up a user in a table, the other value is computed based on the current date and time. In this practice, some factors are not used to provide any functionality. You show their values after you create them to verify that they function properly.

Note: Practice 8-4 is a prerequisite for practice 9-1.

Assumptions

Practices 3-1, 3-2, and 3-4 were successfully completed:

- Database Vault is configured.
- Test users are created.
- The `leo_dvowner` user is configured to have access to Database Vault via Cloud Control.

Practice 8-1: Restricting Access by Using the `Client_IP` and Domain Factors

The Domain factor exists when Database Vault is delivered, but no identities are defined for it. In this practice, you define three Domain identities that connote different levels of trust and access. These can then be used to evaluate whether access can be granted to different areas of the database, under different circumstances.

In this practice, you edit the existing Domain factor and add three identities—**`SECURE`**, **`INTRANET`**, and **`INTERNET`**—to it with trust levels of **Very Trusted**, **Trusted**, and **Untrusted**, respectively. After these identities are added, you create identity mappings for each identity based on the following:

- **`SECURE`**: The `Client_IP` factor has a value equal to the zero-padded IP address of your machine.
- **`INTRANET`**: The `Client_IP` factor has a value where the first three octets of the IP address match your machine, but the final octet does not.
- **`INTERNET`**: The `Client_IP` factor has a value that is *not* like the first three octets of the IP address of your machine.

You then edit the `Non_Work_Hours` rule set to have another rule, called **`Secure`**, that checks for an identity of **`SECURE`** for the Domain factor. This should override the time criteria, allowing a **`SECURE`** connected session to have access whether it is during working hours or not.

1. Modify the existing `Client_IP` factor, changing its retrieval method to zero-pad the IP address. For your convenience, to modify the retrieval method for the **`Client_IP`** factor, use the **`lab_08_01_01.sql`** script. This script forces the last octet of the IP address to be left-padded with zeros to a length of three digits. For example, if the IP address of the machine is **`10.156.49.80`**, the actual string returned by this factor is **`10.156.49.080`**. This is in preparation for the following practices, which make range comparisons of IP addresses.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ cd /home/oracle/labs
$ sqlplus leo_dvowner
```

```
Enter password:
Connected.
SQL> @lab_08_01_01.sql
SQL> set term on
SQL>
SQL>
SQL> -- Change the Client_IP factor to return a zero-padded
string for the last octet.
SQL>
SQL> connect leo_dvowner/oracle_4U
Connected.
SQL> execute dvsys.dbms_macadm.update_factor('Client_IP', -
> 'IP Address', -
```

```
> 'This factor defines the IP Address and retrieval method for a
client to the database server. This version has been updated to
left pad the last octet with zeros to a length of 3 digits.', -
>  NULL, -
>  'substr( UPPER(SYS_CONTEXT(''USERENV'',''IP_ADDRESS'')), 1,
instr(UPPER(SYS_CONTEXT(''USERENV'',''IP_ADDRESS'')),''.'',-
1,1))     || lpad(substr(
UPPER(SYS_CONTEXT(''USERENV'',''IP_ADDRESS'')),
instr(UPPER(SYS_CONTEXT(''USERENV'',''IP_ADDRESS'')),''.'',-1,1)
+ 1), 3, ''0'')', -
>  NULL, -
>  1, -
>  0, -
>  0, -
>  0, -
>  POWER(2,0) -
> );

PL/SQL procedure successfully completed.

SQL>
SQL> commit;

Commit complete.

SQL>
```

2. Test the **`Client_IP`** factor to ensure that it is appropriately set and returns a zero-padded IP address. Connect as the **`HR`** user making sure to specify the service name for the database (forcing a TNS connection). Execute **`select DVF.F$CLIENT_IP from dual`** and verify that the last octet in your IP address is zero-padded to three digits.

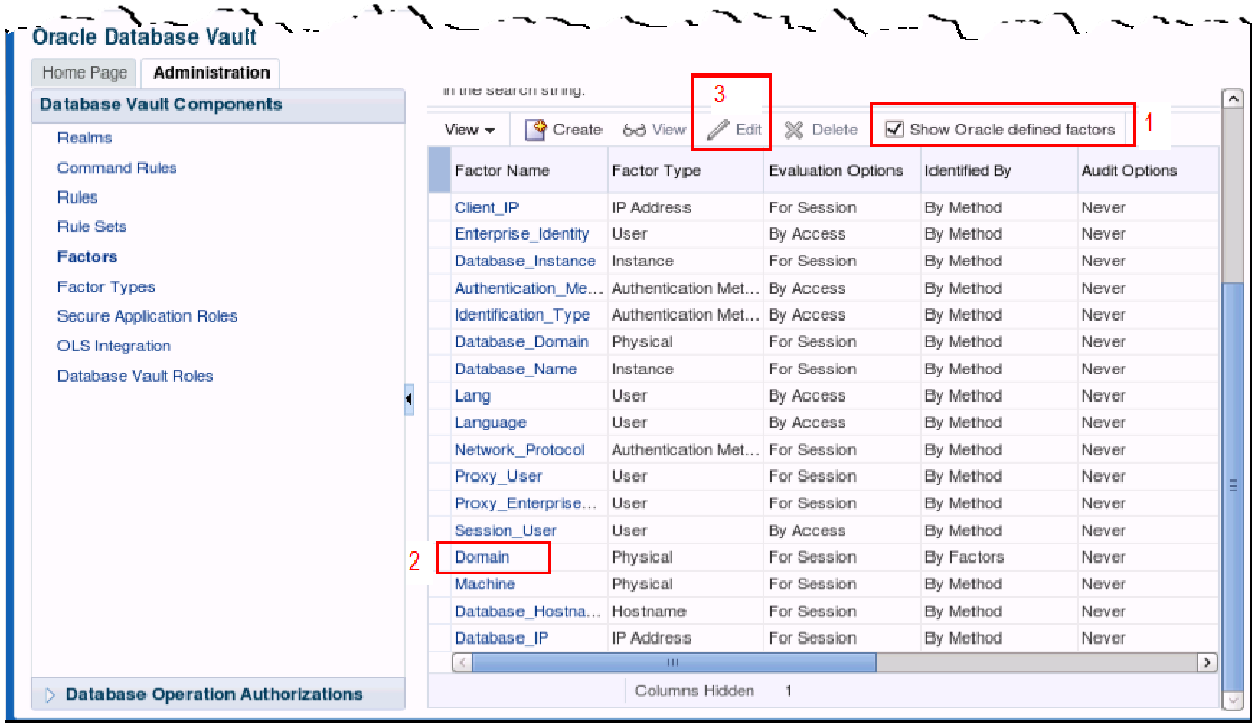
   Note: If the last octet of your IP address is already three digits in length, you will not notice any difference.

```
SQL> connect hr@orcl
Enter password:
Connected.
SQL> select DVF.F$CLIENT_IP from dual;

F$CLIENT_IP
----------------------------------------------------------------
139.185.35.144

SQL>
```

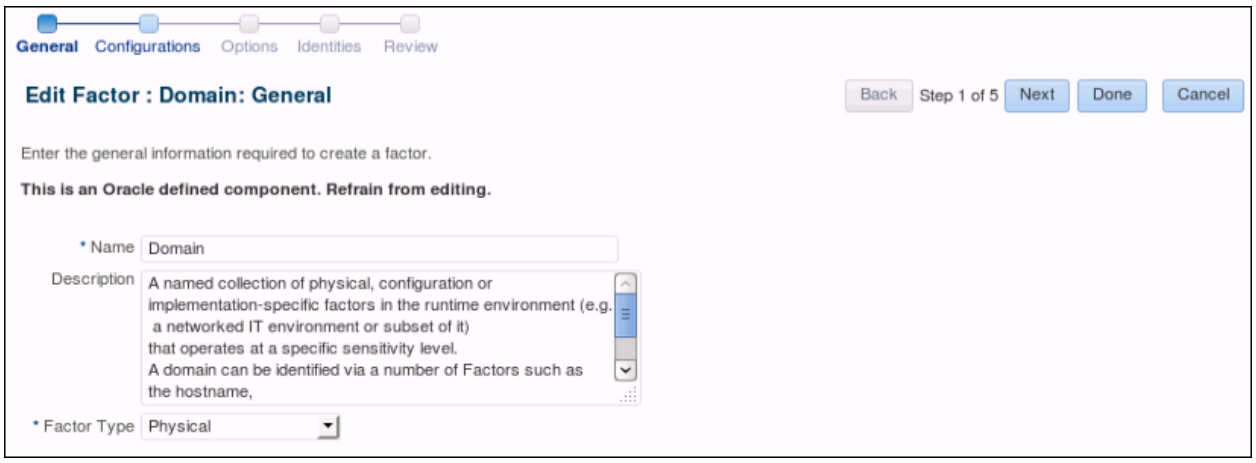
3. Log in to EM Cloud Control as the `leo_dvowner` user. Navigate to Administration (tabbed page) > Factors. Click the "**Show Oracle defined factors**" to display the list of Oracle predefined factors. Edit the Domain factor and add three identities—**`INTERNET`**, **`INTRANET`**, and **`SECURE`**—to it with trust levels of **`Untrusted`**, **`Trusted`**, and **`Very Trusted`** respectively. Use the details in the following table to assist you.

Step	Screen/Page Description	Choices or Values
a.	Factors page	Select the Domain factor and click **Edit**.
b.	Edit Factor: Domain: General	Click **Next**.
c.	Edit Factor: Domain: Configurations	Click **Next**.
d.	Edit Factor: Domain: Options	Click **Next**.
e.	Edit Factor: Domain: Identities	Click **Add New Identity**.
f.	Edit Factor: Domain: Identities Add New Identity	Click **Add New Identity**. Value: **`INTERNET`** Trust Level: **`Untrusted`** Click **OK**.
g.	Edit Factor: Domain: Identities Add New Identity	Click **Add New Identity**. Value: **`INTRANET`** Trust Level: **Trusted** Click **OK**.
h.	Edit Factor: Domain: Identities Add New Identity	Value: **`SECURE`** Trust Level: **Very Trusted** Click **OK**.
i.	Edit Factor: Domain	Review the three identities that you just defined for the Domain factor. Click **Next.**
j.	Edit Factor: Review	Review the entire page, especially the `Show SQL` section. Click **Finish**.
k.	Administration: Factors	You should receive a success message.

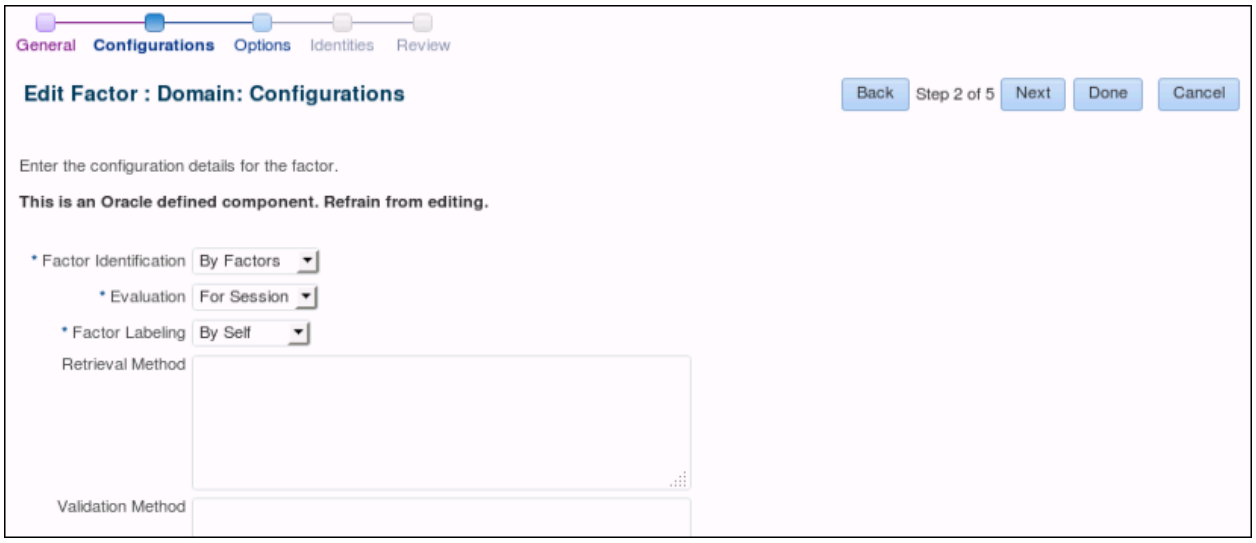
a.



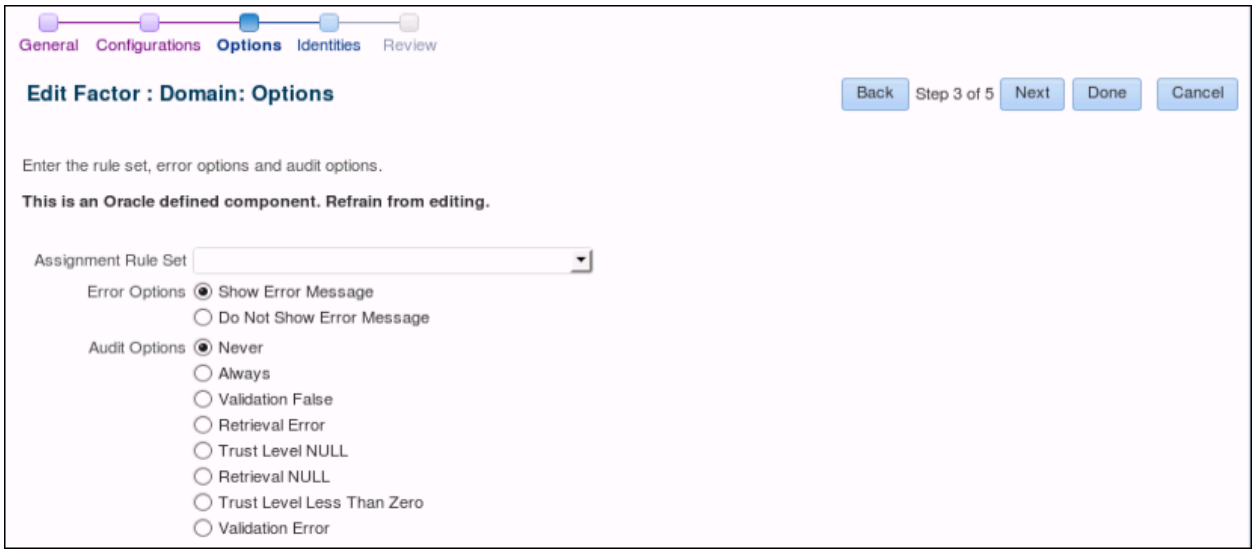
b.



c.

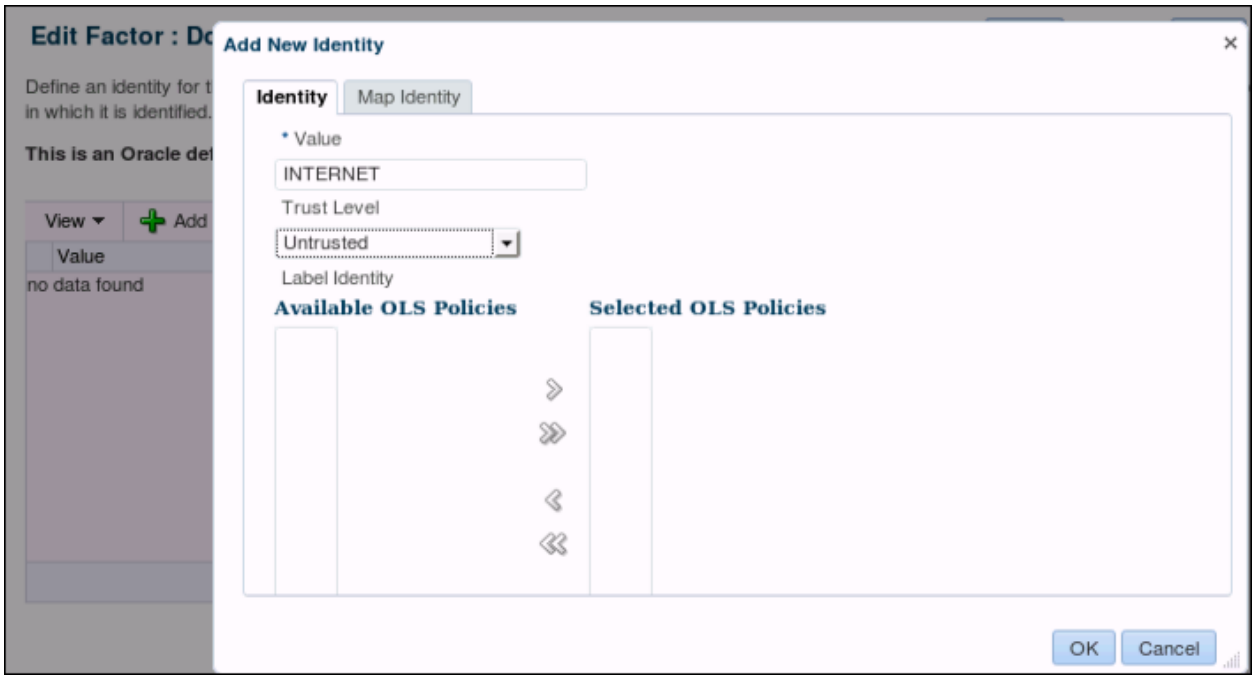


d.

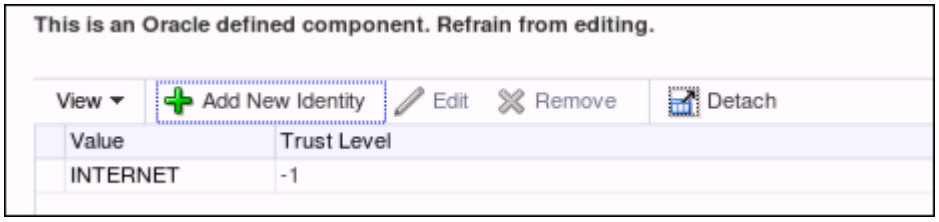


e.

f.

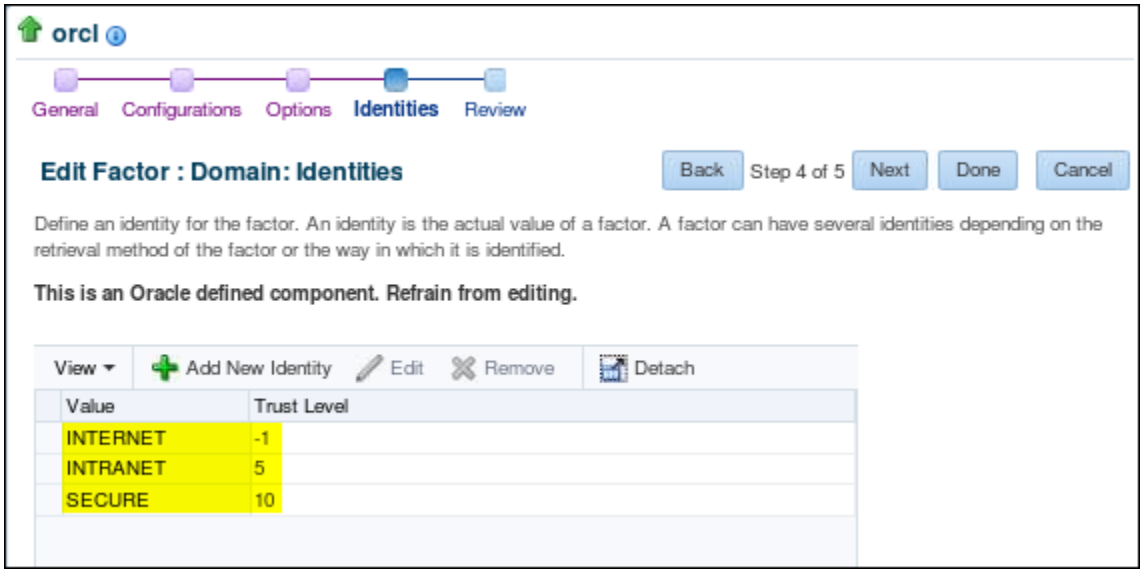


g.

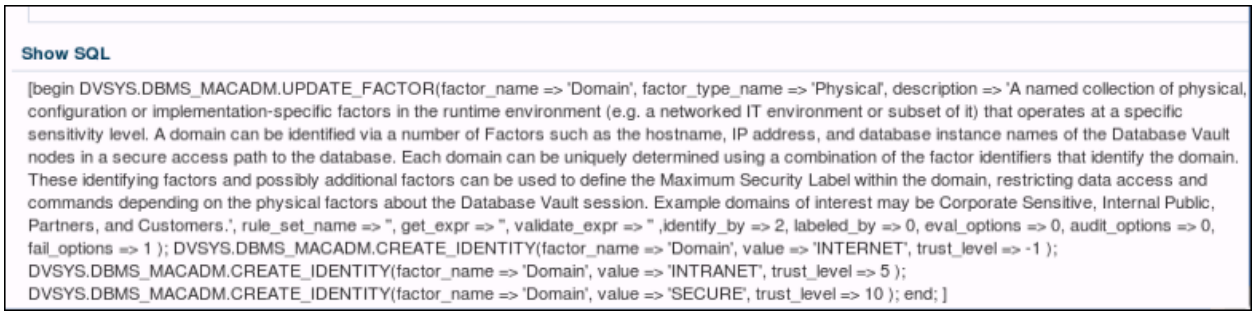


h. No screenshots for this step because it is repetition of the previous ones.

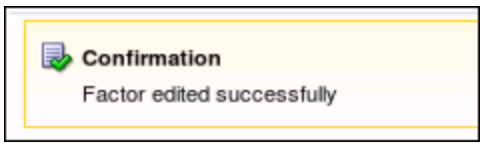
i.



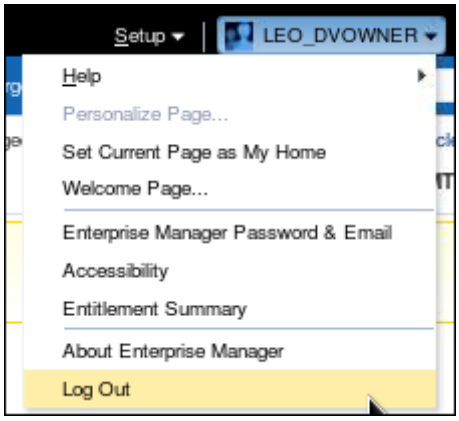
j.

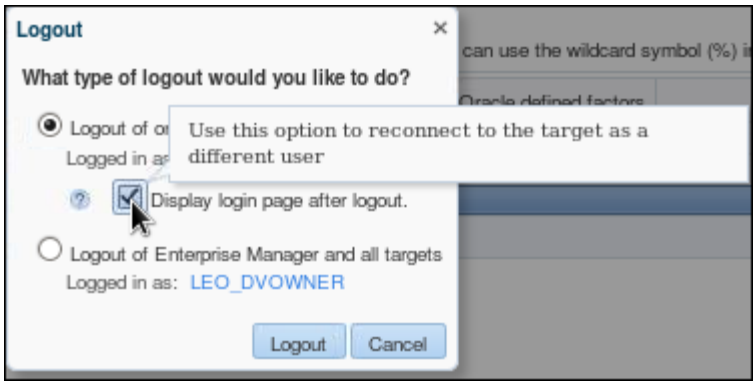


k.



4. Even if you receive the Confirmation message that the factor is edited successfully, this operation needs to be committed. Log out from the `leo_dvowner` session and reconnect as `leo_dvowner`.





5. Verify that the identities are created for the `Domain` factor.

```
SQL> CONNECT leo_dvowner
Enter password:
Connected.
SQL> col FACTOR_NAME format A10
col VALUE format A16
SQL> SQL>
SQL> select FACTOR_NAME, VALUE, TRUST_LEVEL
     from   dvsys.DBA_DV_IDENTITY
     where  factor_name = 'Domain';
  2    3
FACTOR_NAM VALUE              TRUST_LEVEL
---------- ---------------- -----------
Domain     SECURE                    10
Domain     INTRANET                   5
Domain     INTERNET                  -1
SQL>
```

6. Create identity mappings for the three identities.
   Either use the EM Cloud Control **or** SQL*Plus to execute `DVSYS` procedures. The latter method is faster than using EM Cloud Control.

   Method with EM Cloud Control:

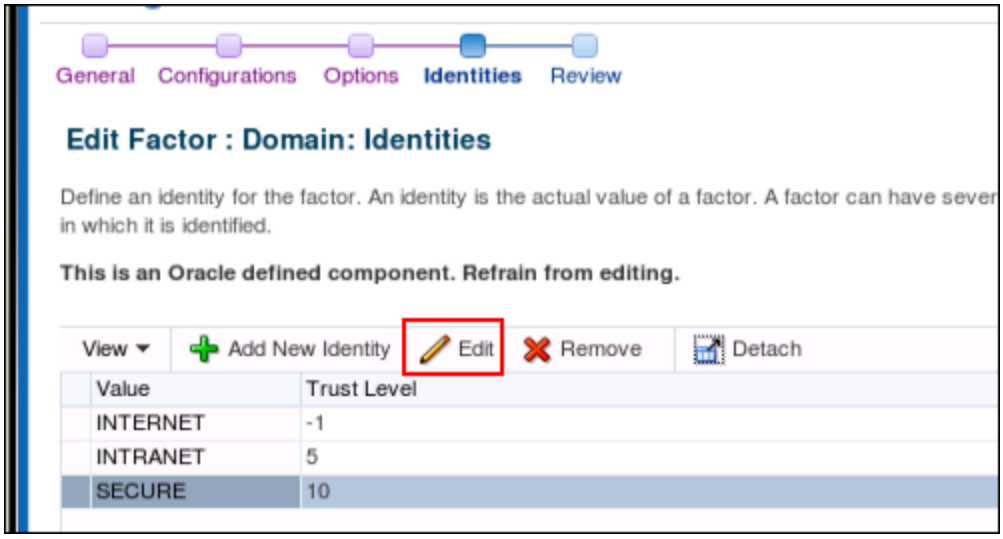
   a. Use the mapping to set the identity to **`SECURE`** when the **`Client_IP`** factor has a value that is equal to the zero-padded IP address of your machine. Use the information in the following table to assist you with this task.
      Navigate to Administration > Factors. Select to edit the Domain factor.

   Hint: You can determine the IP address of your machine by using the **`ifconfig`** OS command or ask your instructor.

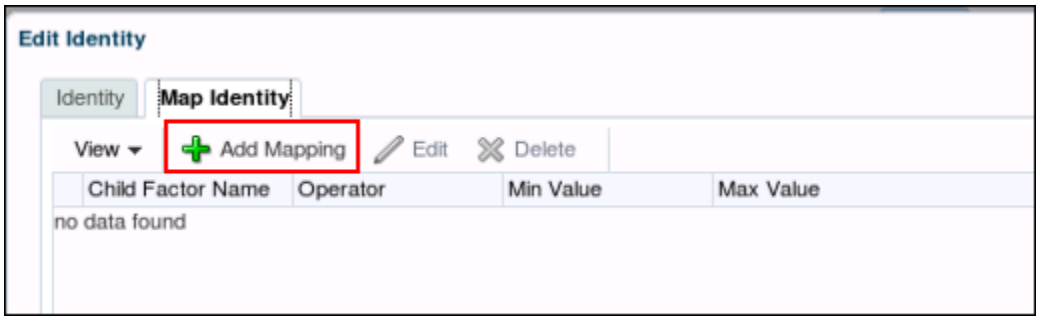
   Note: To avoid many duplicate screenshots, only a few relevant ones are added below.

Step	Screen/Page Description	Choices or Values
1)	Edit Factor: Domain: General	Click **Next**.
2)	Edit Factor: Domain: Configurations	Click **Next**.
3)	Edit Factor: Domain: Options	Click **Next**.
4)	Edit Factor: Domain: Identities	Select the **`SECURE`** identity and click **Edit**.
5)	Edit Identity	Click **Add Mapping** in the Map Identity tab.
6)	Add New Identity Mapping	Child Factor name: **`Client_IP`** Operator: **Equal** Min Value: ***`<your zero-padded IP address>`*** Max Value: Leave it blank. Click **OK**.
7)	Edit Identity	Review the identity that you just mapped and click **OK**.

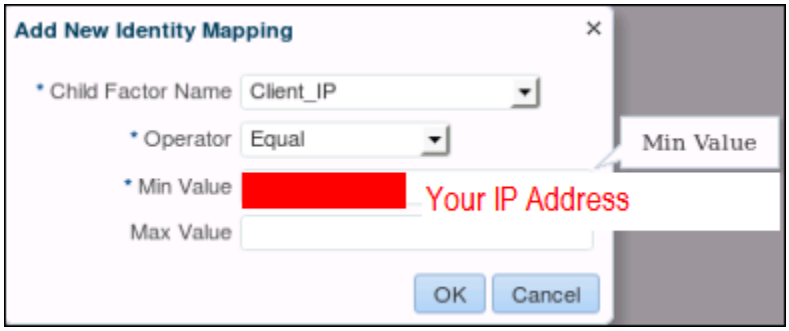
4)



5)



6)



b. Create an identity mapping for the **INTRANET** identity. Use the mapping to set the identity to **INTRANET** when the client machine is on the local subnet, but is not your machine. Because the identities are evaluated in the ASCII sort order of the identity name, **INTRANET** is evaluated first. Because of this, exclude your PC from the set of machines that will be assigned the **INTRANET** identity.

Note: There are no screenshots for the following two sections because they are similar to the previous one.

Step	Screen/Page Description	Choices or Values
1)	Edit Factor: Domain: Identities	Select **INTRANET** identity and click **Edit**.
2)	Edit Identity	Click **Add Mapping** in the Map Identity tab.
3)	Add New Identity Mapping	Child Factor name: **Client_IP** Operator: **Between** Min Value: Your machine's IP address replacing the last octet with **001** Max Value: Your zero-padded machine's IP address replacing the last octet with the number that is 1 less than your machine's last octet value. (For example, if the machine's IP address is `10.156.49.80`, you enter `10.156.49.079`.) Click **OK**.
4)	Edit Identity	Click **Add Mapping** in the Map Identity tab.
5)	Add New Identity Mapping	Child Factor name: **Client_IP** Operator: **Between** Min Value: Your machine's zero-padded IP

Step	Screen/Page Description	Choices or Values
		address adding 1 to the last octet value. (For example, if your machine's IP address is `10.156.49.80`, you enter `10.156.49.081`.) Max Value: Enter your machine's IP address in the High Value field, changing the last octet to **`255`**. Click **OK**.
6)	Edit Identity	Review the identities that you just mapped and click **OK**.

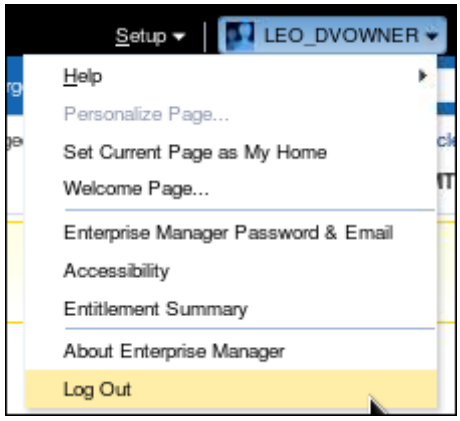
c. Create an identity mapping for the **`INTERNET`** identity. Use the mapping to set the identity to **`INTERNET`** when the client machine is not on the local subnet. In this case, the `Client_IP` factor has a value that is *not* like the first three octets of the IP address of the student PC. For example, if your PC has an IP address of `192.168.1.10`, the local subnet is `192.168.1`. The definition of local subnet can vary depending on the subnet mask defined. The subnet mask for OU classrooms is `255.255.255`, a class C subnet.

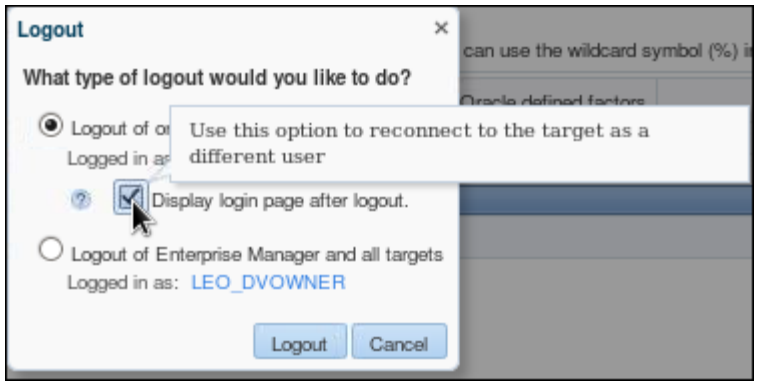
Use the information in the following table to assist you with this task.

Step	Screen/Page Description	Choices or Values
1)	Edit Factor: Domain: Identities	Select **`INTERNET`** identity and click **Edit**.
2)	Edit Identity: INTERNET	Click **Add Mapping** in the Map Identity tab.
3)	Add New Identity Mapping	Child Factor name: **`Client_IP`** Operator: **Not Like** Min Value: IP address of your machine with the last octet replaced with the percentage symbol (for example, **`192.168.1.%`**) max Value: Leave it blank. Click **OK**.
4)	Edit Identity	Review the identity that you just mapped and click **OK**.

d. Validate the Domain Identities completion by clicking **Done** and then **Finish**.

e. Even if you receive the Confirmation message that the factor is edited successfully, this operation needs to be committed. Log out from the `leo_dvowner` session and reconnect as `leo_dvowner`.





f. Logged in to SQL*Plus as the `leo_dvowner` user, verify that the identities and mappings are created.

```
SQL> CONNECT leo_dvowner
Enter password:
Connected.
SQL> col FACTOR_NAME format A6
col IDENTITY_VALUE format A8
col CHILD_FACTOR_NAME format A9
col OPERATION_VALUE format A8
col OPERAND1 format A14
col OPERAND2  format A14
SQL> SQL> SQL> SQL> SQL>
SQL> select FACTOR_NAME, IDENTITY_VALUE, CHILD_FACTOR_NAME,
            OPERATION_VALUE, OPERAND1, OPERAND2
     from dvsys.DBA_DV_IDENTITY_MAP order by 2, 5;

FACTOR IDENTITY CHILD_FAC OPERATIO OPERAND1
------ -------- --------- -------- --------------
OPERAND2
--------------
Domain INTERNET Client_IP Not Like 139.185.35.%
Domain INTRANET Client_IP Between  139.185.35.001 139.185.35.143
```

```
Domain INTRANET Client_IP Between  139.185.35.145 139.185.35.255
Domain SECURE   Client_IP Equal    139.185.35.144

SQL>
```

Method with SQL*Plus:

*(If you used the Cloud Control method, then you must skip the SQL*Plus method because the task is already done.)*

Replace the operands with YOUR IP address information.

a. Logged in to SQL*Plus as the `leo_dvowner` user, execute the following **`DVSYS.DBMS_MACADM`** procedures.

```
SQL> CONNECT leo_dvowner
Enter password:
Connected.
SQL> exec DVSYS.DBMS_MACADM.ADD_FACTOR_LINK(-
  parent_factor_name => 'Domain', -
  child_factor_name => 'Client_IP', -
  label_indicator => 'Y')
> > >

PL/SQL procedure successfully completed.

SQL> exec DVSYS.DBMS_MACADM.CREATE_IDENTITY_MAP(-
  identity_factor_name  => 'Domain', -
  identity_factor_value => 'INTERNET', -
  parent_factor_name    => 'Domain', -
  child_factor_name     => 'Client_IP', -
  operation             => 'NOT LIKE', -
  operand1              => '139.185.35.%' ,-    <<< Your IP info
  operand2              => '' )
> > > > > > >

PL/SQL procedure successfully completed.

SQL> exec DVSYS.DBMS_MACADM.CREATE_IDENTITY_MAP(-
  identity_factor_name  => 'Domain', -
  identity_factor_value => 'INTRANET', -
  parent_factor_name    => 'Domain', -
  child_factor_name     => 'Client_IP', -
  operation             => 'BETWEEN', -
  operand1              => '139.185.35.001' ,-   <<< Your IP
info
  operand2              => '139.185.35.143' )    <<< Your IP
info
```

```
> > > > > > > >

PL/SQL procedure successfully completed.

SQL> exec DVSYS.DBMS_MACADM.CREATE_IDENTITY_MAP(-
  identity_factor_name  => 'Domain', -
  identity_factor_value => 'INTRANET', -
  parent_factor_name    => 'Domain', -
  child_factor_name     => 'Client_IP',-
  operation             => 'BETWEEN', -
  operand1              => '139.185.35.145' ,-   <<< Your IP
info
  operand2              => '139.185.35.255' )    <<< Your IP
info
> > > > > > > >

PL/SQL procedure successfully completed.

SQL> exec DVSYS.DBMS_MACADM.CREATE_IDENTITY_MAP(-
  identity_factor_name  => 'Domain', -
  identity_factor_value => 'SECURE', -
  parent_factor_name    => 'Domain', -
  child_factor_name     => 'Client_IP', -
  operation             => '=', -
  operand1              => '139.185.35.144' ,-  <<< Your IP info
  operand2              => '' )
> > > > > > > >

PL/SQL procedure successfully completed.
SQL>
```

b. In the same session, verify the definitions of the identities and mappings, and then COMMIT.

```
SQL> col FACTOR_NAME format A6
col IDENTITY_VALUE format A8
col CHILD_FACTOR_NAME format A9
col OPERATION_VALUE format A8
col OPERAND1 format A14
col OPERAND2  format A14
SQL> SQL> SQL> SQL> SQL>
SQL> select FACTOR_NAME, IDENTITY_VALUE, CHILD_FACTOR_NAME,
            OPERATION_VALUE, OPERAND1, OPERAND2
     from dvsys.DBA_DV_IDENTITY_MAP order by 2, 5;
```

```
FACTOR IDENTITY CHILD_FAC OPERATIO OPERAND1
------ -------- --------- -------- --------------
OPERAND2
-------------
Domain INTERNET Client_IP Not Like 139.185.35.%
Domain INTRANET Client_IP Between  139.185.35.001 139.185.35.143
Domain INTRANET Client_IP Between  139.185.35.145 139.185.35.255
Domain SECURE   Client_IP Equal    139.185.35.144

SQL>

SQL> COMMIT;

Commit complete.

SQL>
```

7. Test the Domain factor identities.
   - Connect to your machine by using the service name and display the identity of the Domain factor.
   - Connect to your machine without using a service name and display the identity of the Domain factor.
   - *The following test is only possible in a classroom with more than one machine.* Connect to another machine in the classroom by using `ssh` and display the identity again. They should be different. This step requires the use of network services. Create a network service name for your database following the pattern **`orcl_<machine name>`**. The name of the machine is found by using the `hostname` command. The instructor tells you how to set the network service name.

   a. Connect to the database by using SQL*Plus, using a network service name for your machine. Verify that the identity of the Domain factor is **`SECURE`**.

```
$ sqlplus hr@orcl
Enter password:
SQL> select DVF.F$DOMAIN from dual;

F$DOMAIN
------------------------------------------------------------
SECURE

SQL>
```

   b. Connect to the database by using SQL*Plus, but without using a service name. What is the identity of the Domain factor in this case? Why?

```
SQL> connect hr
Enter password:
Connected.
```

```
SQL> select DVF.F$DOMAIN from dual;

F$DOMAIN
------------------------------------------------------------
INTERNET

SQL> EXIT
$
```

The answer is that the identity is `NULL`. This is because when you log in without a service name, the IP address is not set for the session. This meets the condition for the `INTERNET` factor because NULL is not like the subnet string being evaluated. Note that the order does not matter in this case because this is the only mapping of the three that has a satisfactory condition.

c. *The following test is only possible in a classroom with more than one machine.* Connect to your database by using a SQL*Plus session that originates from a different machine in the classroom. Use the **`ssh`** utility to start a session on the instructor machine. (The instructor may provide an alternate machine or IP address for this purpose.) Verify that Domain is set to **`INTRANET`**. You can use the following commands to perform this.

Note: Remember to disconnect from the `ssh` session before continuing with the practice.

```
$ ssh <your_neighbor_hostname>
The authenticity of host 'edrsr10p1 (139.185.35.110)' can't be
established.
RSA key fingerprint is
c3:63:4f:f5:0c:44:e8:a8:af:a9:44:da:e7:1f:c4:84.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'edrsr10p1,139.185.35.110' (RSA) to
the list of known hosts.
oracle@edrsr10p1's password:
Last login: Thu Jun 12 06:32:06 2014 from 141.144.18.156
$ . oraenv
ORACLE_SID = [oracle] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus hr@orcl_<your_neighbor_hostname>

Enter password:

SQL> select DVF.F$DOMAIN from dual;

F$DOMAIN
------------------------------------------------------------
```

```
INTRANET

SQL> EXIT
$ exit
logout
Connection to <your_neighbor_hostname> closed.
$
```

8. Logged in to Cloud Control as the `leo_dvowner` user, edit the `Non_Work_Hours` rule set to have another rule, called **`Secure`**, that checks for an identity of **`SECURE`** for the Domain factor. This will override the time criteria, allowing a **`SECURE`** connected session to have access whether it is during working hours or not. To get started, navigate to the Administration tabbed page and click **Rule Sets**. Refer to the information in the following table.

   Note: Screenshots are not included because they have been previously shown.

Step	Screen/Page Description	Choices or Values
a.	Rule Sets page	Select `Non_Work_Hours` and click **Edit**.
b.	Edit Rule Set: Non Work Hours: General	Click **Next**.
c.	Edit Rule Set: Non Work Hours: Associate with Rules	Click **Create Rule**.
d.	Create Rule	Name: **`Secure`** Expression: **`dvf.f$domain = 'SECURE'`** Click **OK**.
e.	Edit Rule Set: Non Work Hours	Review the rules in the rule set and click **Next.`**
f.	Edit Rule Set : Non_Work_Hours: Error Handling and Audit Options	Click **Next**.
g.	Edit Rule Set : Non_Work_Hours: Review	Review the entire rule sets page, especially the code at the bottom, then click **Finish**.

9. Return to SQL*Plus and reconnect as **`BERNST`** by using a net service name. Then reattempt to create the table in the **`HR`** schema as the **`BERNST`** user. What happens, and why?

```
$ sqlplus bernst@orcl
Password:
Connected.
SQL> create table hr.x (a number);

Table created.

SQL> drop table hr.x;

Table dropped.

SQL>
```

Note: If the table exists, just drop it.

The statement succeeds now during work hours, because the alternative requirement of having the **SECURE** identity for the Domain factor satisfies the authorization rule set. (Remember: The evaluation option is set to `Any True`, requiring that only one of the three rules be true).

Practice 8-2: Creating a Factor to Determine Job Role

In this practice, you create a function to return `JOB_ID` from the `EMPLOYEES` table for the connected user if his or her username matches the value in the `EMAIL` column. You then create a factor that uses this function to determine a user's job role. Test that the factor functions properly by using the Database Vault API in SQL*Plus.

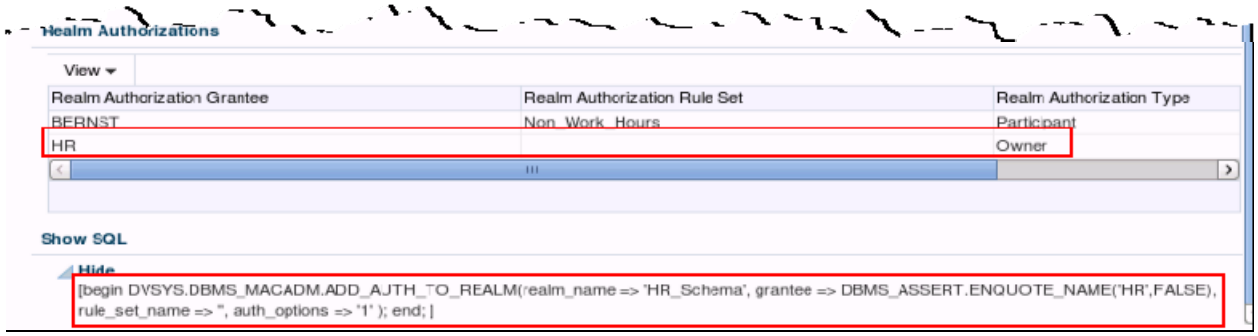
Note: In this practice, you create a factor and test its functionality by using SQL*Plus. You will not use this factor to provide further functionality.

1. In this practice, you need to create a function in the `HR` schema, and you decide to authorize the `HR` user as an owner on the `HR` Schema realm. Edit the **`HR Schema`** realm and add the **`HR`** user as an owner. Still logged in to EM Cloud Control as the `leo_dvowner` user, navigate to Administration (tabbed page) > Realms. Use the details provided in the following table to assist you.

   Note: To avoid redundancy, previously shown screenshots are not included.

Step	Screen/Page Description	Choices or Values
a.	Realms	Select HR Schema and click **Edit**.
b.	Edit Realm: `HR` Schema: General	Click **Next**.
c.	Edit Realm: `HR` Schema: Realm Secured Objects	Click **Next**.
d.	Edit Realm: `HR` Schema: Realm Authorizations	Click **Add**. Realm Authorization Grantee: **`HR [USER]`** Realm Authorization Type: **Owner** Realm Authorization Rule Set: **`<Non Selected>`** Click **OK**. Click **Next.**
e.	Edit Realm: `HR` Schema: Review	Review your modifications, especially the code on the bottom of the page. Click **Finish** to finish editing the realm.
f.	Realms	You should receive a success message.

e.



2. Create a function called **`GET_JOB_ID`** in the **`HR`** schema. This function returns **`JOB_ID`** of the connected user if the user login matches the **`EMAIL`** column of the **`HR.EMPLOYEES`** table. Connect as the **`HR`** user and run the **`get_job_id.pls`** script to create this function.

   Note: The password for the **`HR`** user is **`oracle_4U`**.

```
SQL> connect hr
```

```
Enter password:
Connected.
SQL> CREATE OR REPLACE FUNCTION   HR.GET_JOB_ID
RETURN VARCHAR2 AS
  v_job_id VARCHAR2(10);
BEGIN
  SELECT job_id INTO v_job_id FROM HR.EMPLOYEES
  WHERE EMAIL = DVF.F$SESSION_USER;
  RETURN v_job_id;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RETURN NULL;
END;
/
  2    3    4    5    6    7    8    9   10   11   12
Function created.

SQL>
```

3. If you want to create a factor that uses this new **GET_JOB_ID** function, you must grant, as the **HR** user, the **EXECUTE** privilege on this function to the **DVSYS** user.

```
SQL> grant execute on get_job_id to DVSYS;

Grant succeeded.
SQL>
```

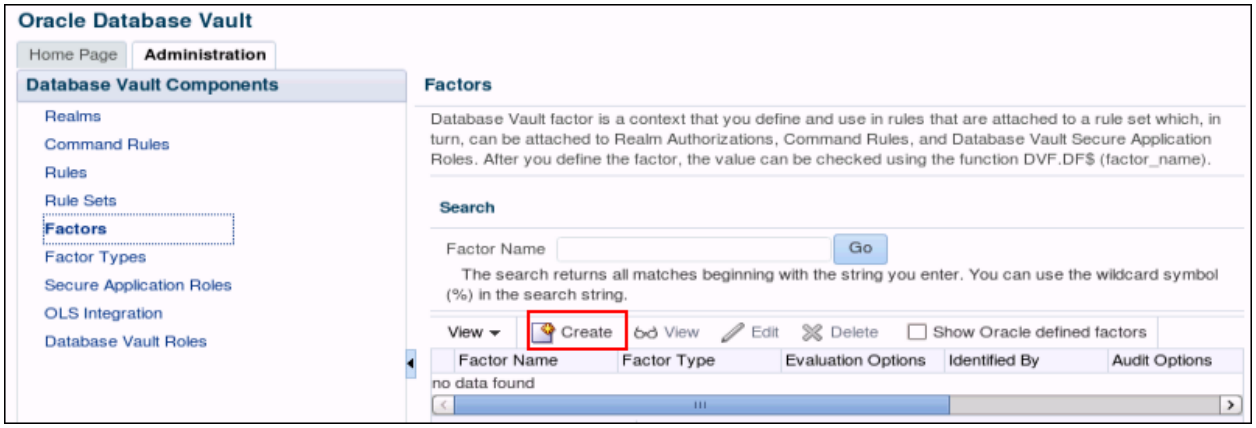
4. Create a factor called **JOBROLE**. Note that all users in these practices appear in the **HR.EMPLOYEES** table. The **JOBROLE** factor uses the user ID, matches it to a row in the **EMPLOYEES** table, and determines **JOBROLE** from the **JOB_ID** column. The factor retrieval method is **HR.GET_JOB_ID()**. Accept the default values for the other settings. Still logged in to EM Cloud Control as the `leo_dvowner` user, navigate to Administration (tabbed page) > Factors. Use the information in the following table for assistance.

   Note: This is the first time that you are creating a factor, so all screenshots are included.

Step	Screen/Page Description	Choices or Values
a.	Factors	Click **Create**.
b.	Create Factor: General	Name: **JOBROLE** Description: **Check the JOB_ID of the connected user.** Factor Type: **Application** Click **Next**.
c.	Create Factor: Configurations	Factor Identification: **By Method** Evaluation: **For Session** Factor Labeling: **By Self** Retrieval Method: **HR.GET_JOB_ID()**

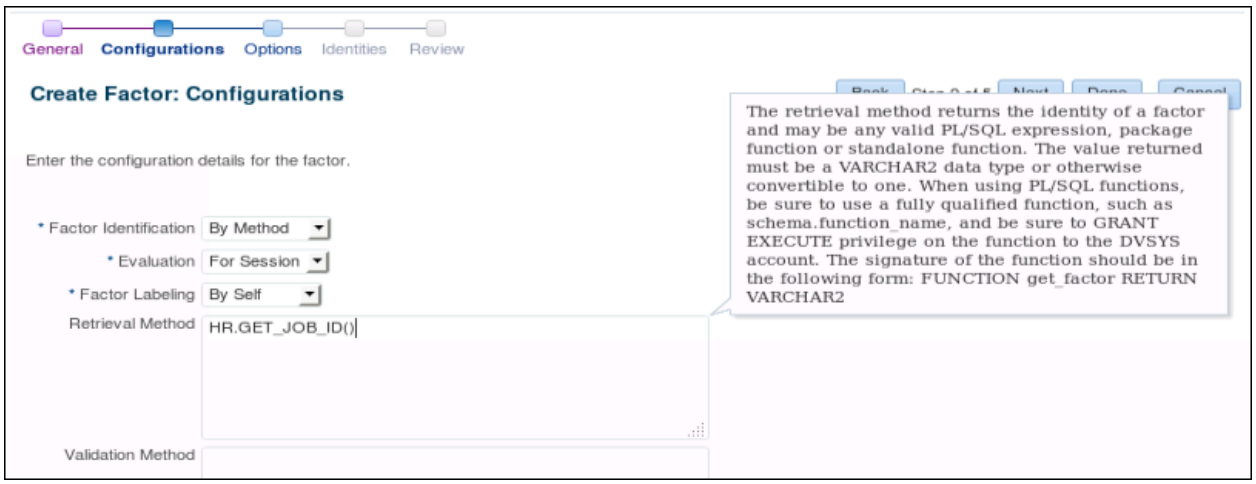
Step	Screen/Page Description	Choices or Values
		Click **Done**.
d.	Create Factor: Review	Review your specifications, especially the code on the bottom of the page. Click **Finish**.
e.	Factors	You should receive a success message.

a.



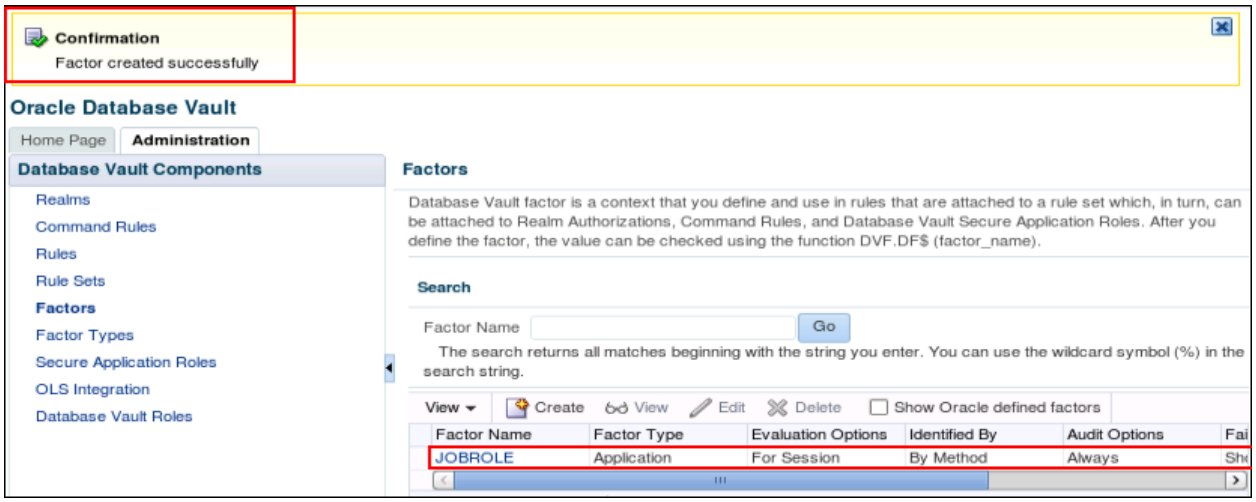
b.

c.



d.

e.



5. Now that you have created the factor, verify that the factor is set properly by using SQL*Plus. This can be done in one of two ways: by using the Database Vault API **`GET FACTOR`** function or by executing **`SELECT DVF.F$JOBROLE FROM DUAL`**. The solutions that follow use the second option.

   Log in to SQL*Plus as **`SMAVRIS`** and execute the `SELECT` statement shown. Next, connect to **`KPARTNER`** and execute the same `SELECT` statement. Finally, connect as the **`HR`** user and execute the same `SELECT` statement.

Question: Why are the values different for each user? Why are the results for the **HR** user NULL?

```
SQL> connect smavris
Enter password:
Connected.
SQL> select dvf.F$JOBROLE from dual;

F$JOBROLE
------------------------------------------------------------
HR_REP

SQL> connect kpartner
Enter password:
Connected.
SQL> select dvf.F$JOBROLE from dual;

F$JOBROLE
------------------------------------------------------------
SA_MAN

SQL> connect hr
Enter password:
Connected.
SQL> select dvf.F$JOBROLE from dual;

F$JOBROLE
------------------------------------------------------------

SQL>
```

Answer: The values are different based on the data in the **HR.EMPLOYEES** table for each user that exists in the table. The **HR** user is not listed in the **HR.EMPLOYEES** table. Therefore, a NULL value is displayed for the **HR** user.

Practice 8-3: Using Assignment Rule Sets with Factors

In this practice, you set an assignment rule set for a factor and see the results.

Assumptions

Practices 3-1, 3-2, and 3-4 were successfully completed.

1. Observe the **JOBROLE** factor value for **BERNST**, **SMAVRIS**, and **WSMITH**.
   a. Connect to **BERNST**. What is the **JOBROLE** factor value?

```
SQL> connect bernst
Password:
Connected.
SQL> SELECT DVF.F$JOBROLE from DUAL;

F$JOBROLE
------------------------------------------------------------
IT_PROG

SQL>
```

   b. Connect to **SMAVRIS**. What is the **JOBROLE** factor value?

```
SQL> connect smavris
Password:
Connected.
SQL> SELECT DVF.F$JOBROLE from DUAL;

F$JOBROLE
------------------------------------------------------------
HR_REP

SQL>
```

   c. Connect to **WSMITH**. What is the **JOBROLE** factor value?

```
SQL> connect WSMITH
Password:
Connected.
SQL> SELECT DVF.F$JOBROLE from DUAL;

F$JOBROLE
------------------------------------------------------------
SA_REP

SQL>
```

2. Try to change the **JOBROLE** for **WSMITH** by using the **DVSYS.SET_FACTOR** procedure. Attempt to set the **JOBROLE** factor to **IT_PROG**.

   Question: What happens, and why?

```
SQL> execute dvsys.set_factor('JOBROLE', 'IT_PROG')
BEGIN dvsys.set_factor('JOBROLE', 'IT_PROG'); END;

*
ERROR at line 1:
ORA-47392: Factor JOBROLE cannot be set
ORA-06512: at "DVSYS.DBMS_MACSEC", line 3
ORA-06512: at "DVSYS.DBMS_MACSEC", line 68
ORA-06512: at "DVSYS.SET_FACTOR", line 5
ORA-06512: at line 1

SQL>
```

   Answer: The **SET_FACTOR** call fails because there is no assignment rule set defined for the **JOBROLE** factor.

3. Add an assignment rule set to the **JOBROLE** factor. The rule set should always be true. Use the delivered Enabled rule set.
   Still logged in to EM Cloud Control as the `leo_dvowner` user, navigate to Administration (tabbed page) > Factors. Use the information in the following table for assistance.

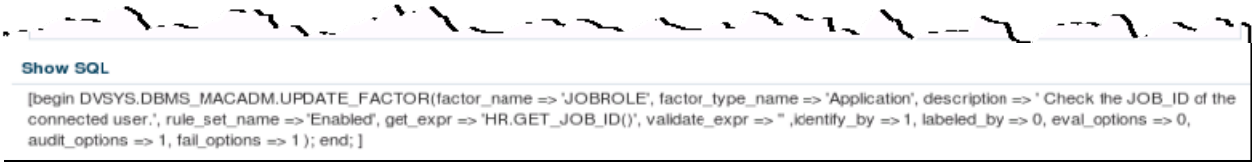
   Note: See relevant screenshots below.

Step	Screen/Page Description	Choices or Values
a.	Factors	Select **JOBROLE**. Click **Edit**.
b.	Edit Factor: JOBROLE: General	Click **Next**.
c.	Edit Factor: JOBROLE: Configurations	Click **Next**.
d.	Edit Factor: JOBROLE: Options	Assignment Rule Set: **Enabled** Click **Done**.
e.	Edit Factor: JOBROLE: Review	Review the code. Click **Finish**.
f.	Factors	You should receive a success message.

   Note: The Enabled rule set always evaluates to true.

d.

e.



4. Reattempt to set the factor as the **WSMITH** user. What happens, and why?

```
SQL> exec dvsys.set_factor('JOBROLE','IT_PROG')

PL/SQL procedure successfully completed.

SQL>
```

You are able to set the **JOBROLE** factor because it now has an assignment rule set that evaluates to true.

5. As the **WSMITH** user, verify that the **JOBROLE** factor was modified.

```
SQL> select dvf.f$jobrole from dual;

F$JOBROLE
-------------------------
IT_PROG

SQL>
```

6. Set the factor to **'xyz'** for **WSMITH** and select it to see that it is set.

```
SQL> exec dvsys.set_factor('JOBROLE','xyz')

PL/SQL procedure successfully completed.

SQL> select dvf.f$jobrole from dual;

F$JOBROLE
--------------------------------------------------------------
xyz


SQL>
```

7. What can you deduce about the ability to set a factor value?

   Answer:

   All that is necessary is that the factor has an assignment rule set defined for it (making it assignable) and that the rule set evaluates to true. Then the factor can be set to any value, regardless of its other attributes or associated identities.

Practice 8-4: Using Rule Sets to Restrict Connection Sources

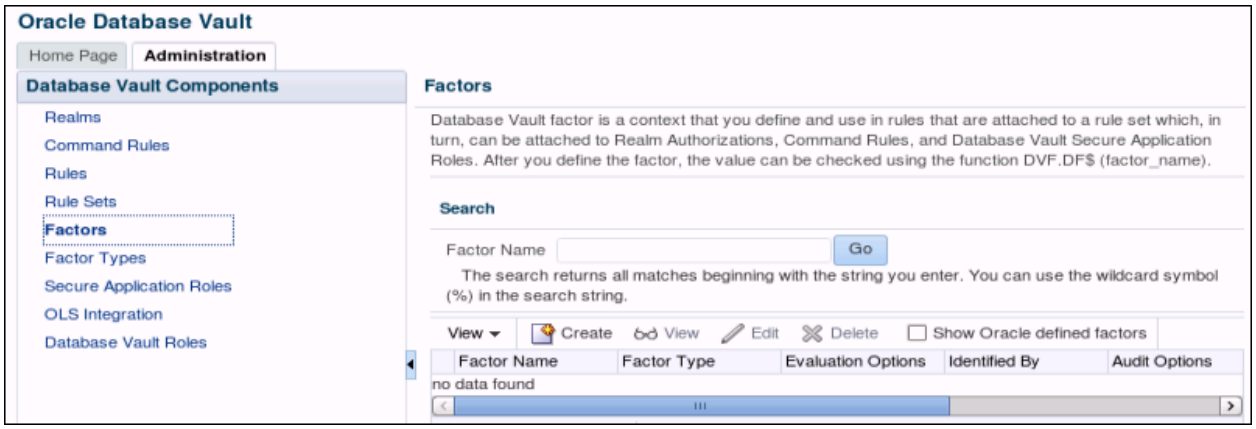
The JOBROLE factor is now settable, but it is made settable by an always-true rule set. In this exercise, you change the assignment rule set to be something more meaningful, requiring that the session have a specific context setup and a specific role active.

Note: This practice is a prerequisite for practice 9-1.

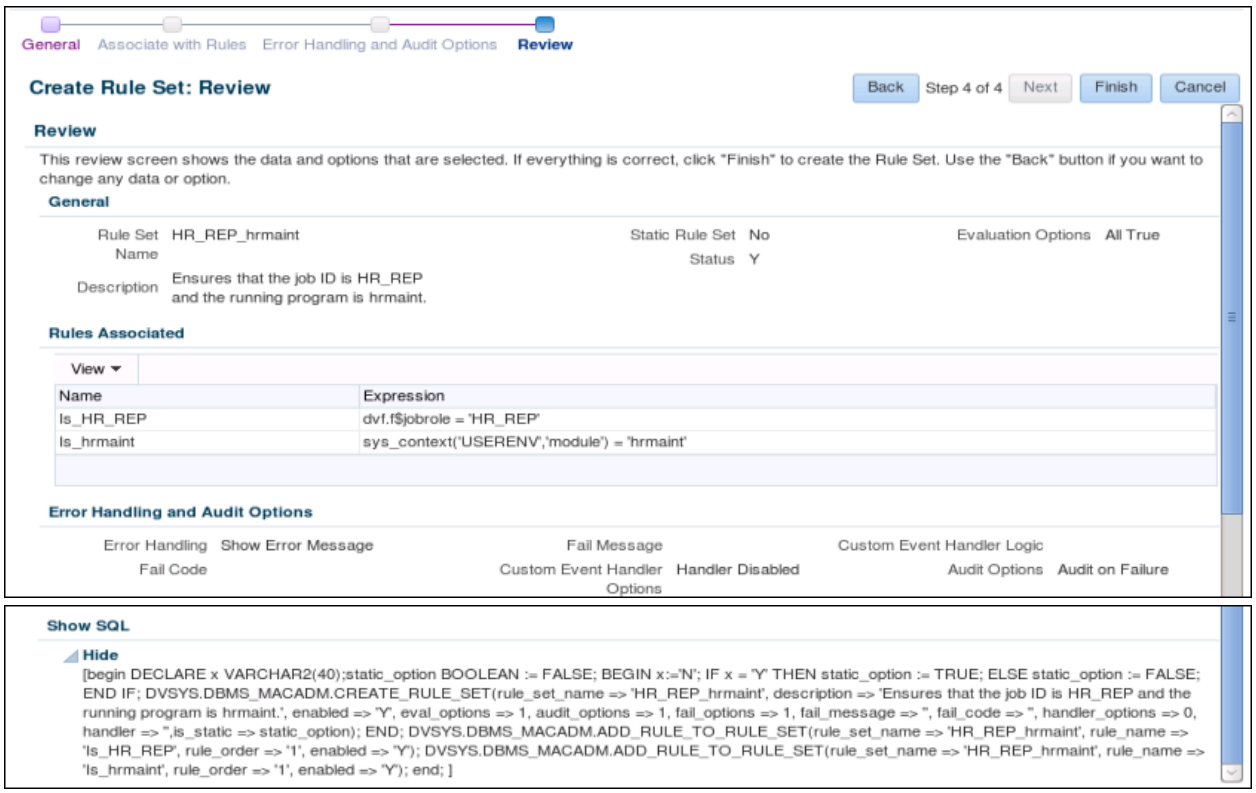
1. Create the **HR_REP_hrmaint** rule set that ensures that the job ID is **HR_REP** and the program that is running is **hrmaint**.
   Still logged in to EM Cloud Control as the leo_dvowner user, first navigate to Administration (tabbed page) > Rule Sets. Refer to the information in the following table.

Step	Screen/Page Description	Choices or Values
a.	Rule Sets page	Click **Create**.
b.	Create Rule Set: General	Rule Set Name: **HR_REP_hrmaint** Description: **Ensures that the job ID is HR_REP and the running program is hrmaint.** Click **Next**.
c.	Create Rule Set: Associate with Rules	Click **Create Rule**.
d.	Create Rule	Name: **Is_HR_REP** Expression: **dvf.f$jobrole = 'HR_REP'** Click **OK**.
e.	Create Rule Set: Associate with Rules	Click **Create Rule**.
f.	Create Rule	Name: **Is_hrmaint** Expression: **sys_context('USERENV','module') = 'hrmaint'** Click **OK**.
g.	Create Rule Set: Associate with Rules	Click **Done** to finish editing the rules in the rule set.
h.	Create Rule Set: Review	Review the rule set that you created and click **Finish**.
i.	Rule Sets	You should receive a success message.

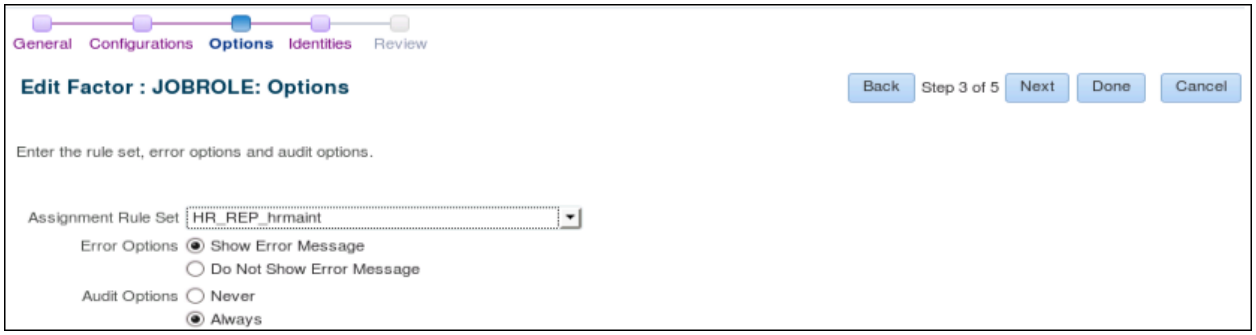
f.



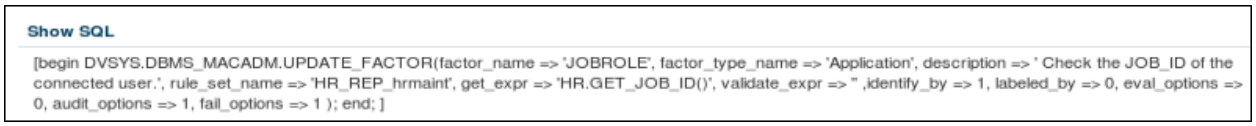
h.



2. Test the new rule set. Make it the assignment rule set for the **JOBROLE** factor and attempt to execute the same **SET_FACTOR** call as was done in the previous practice. The **SET_FACTOR** call should now fail. When you finish testing, set the factor's assignment rule set back to its original setting.
   a. Logged in to Cloud Control as the `lea_dvowner` user, navigate to the Factors page, select the **JOBROLE** factor, and click **Edit**.
   b. On the Edit Factor: JOBROLE: Options page, set the Assignment Rule Set to **HR_REP_hrmaint**, and click **Done**.



c. Review the code and click **Finish**.

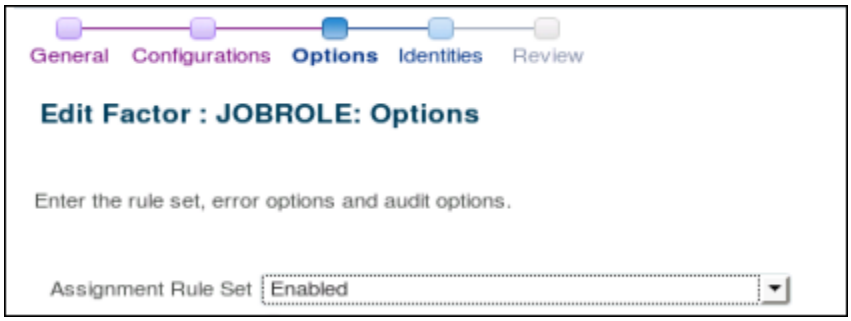


d. Attempt to set the factor from a SQL*Plus session, using the **WSMITH** user.
Note: If you are continuing with the `WSMITH` session that you had used for the steps at the end of Practice 8-3, you will still see the **JOBROLE** factor as `xyz`. The factor evaluation is `For Session`. Therefore, reconnect so that the factor is reevaluated.

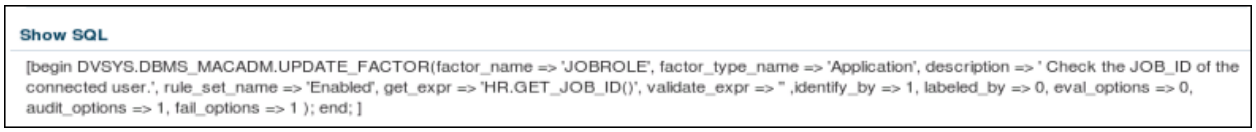
```
SQL> connect WSMITH
Password:
Connected.
SQL> SELECT DVF.F$JOBROLE from DUAL;

F$JOBROLE
------------------------------------------------------------
SA_REP

SQL> exec dvsys.set_factor('JOBROLE','IT_PROG')
BEGIN dvsys.set_factor('JOBROLE','IT_PROG'); END;

*
ERROR at line 1:
ORA-47391: attempt to set Factor JOBROLE violates Rule Set
HR_REP_hrmaint
ORA-06512: at "DVSYS.DBMS_MACSEC", line 3
ORA-06512: at "DVSYS.DBMS_MACSEC", line 68
ORA-06512: at "DVSYS.SET_FACTOR", line 5
ORA-06512: at line 1

SQL>
```

3. To prepare for future practices, return to the **JOBROLE** factor in Administration (tabbed page) > Factors and set the Assignment Rule back to Enabled.
   a. On the Factors page, select the **JOBROLE** factor and click **Edit**.

b. Set the Assignment Rule to **`Enabled`** and click **Done**.



c. Review the page, especially the code at the bottom, and click **Finish**.



Practice 8-5: Using a Factor to Identify a User

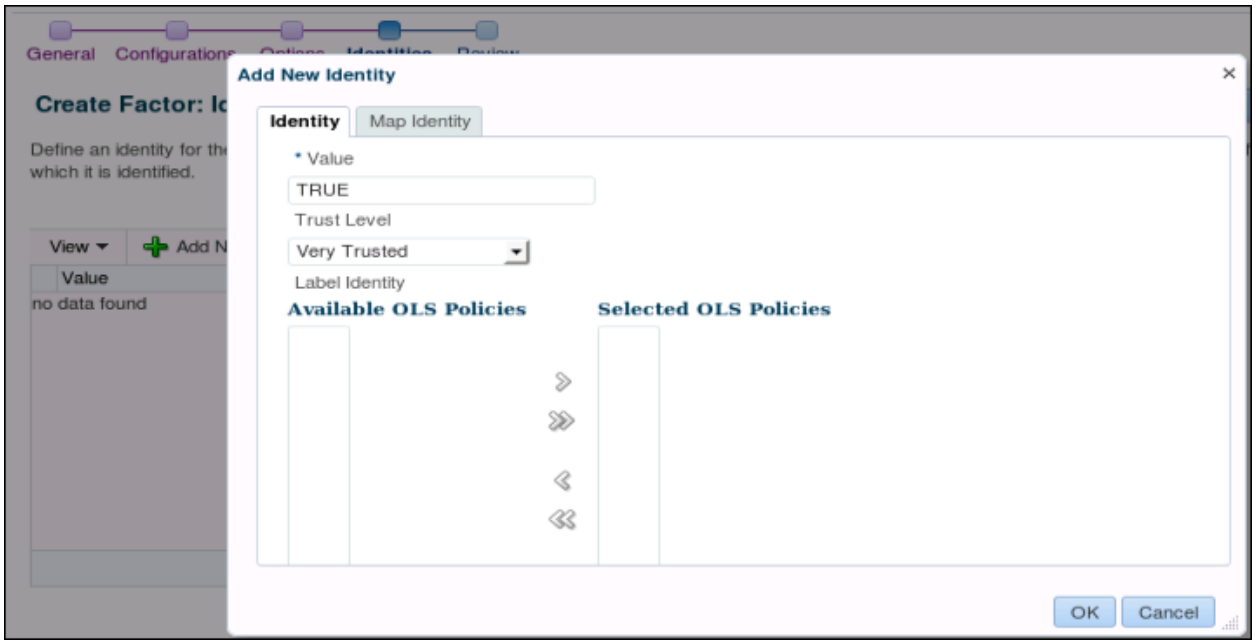
In this practice, you create a factor called **IS_HR_DBA**. This factor can be used to identify a user as the **HR_Schema** realm owner. Set the Identification to By Factors, and create an identity of **TRUE**. Create an identity mapping that sets the identity to **TRUE** when the Session_user factor is equal to **BERNST**, the current HR application DBA. Test the factor by connecting to both **BERNST** and another user (such as **HR**), and selecting the value of the **IS_HR_DBA** factor. The value should be **TRUE** for **BERNST** and null for any other user.

1. Navigate to the Factors page and click **Create**. Use the information in the following table to create the **IS_HR_DBA** factor.

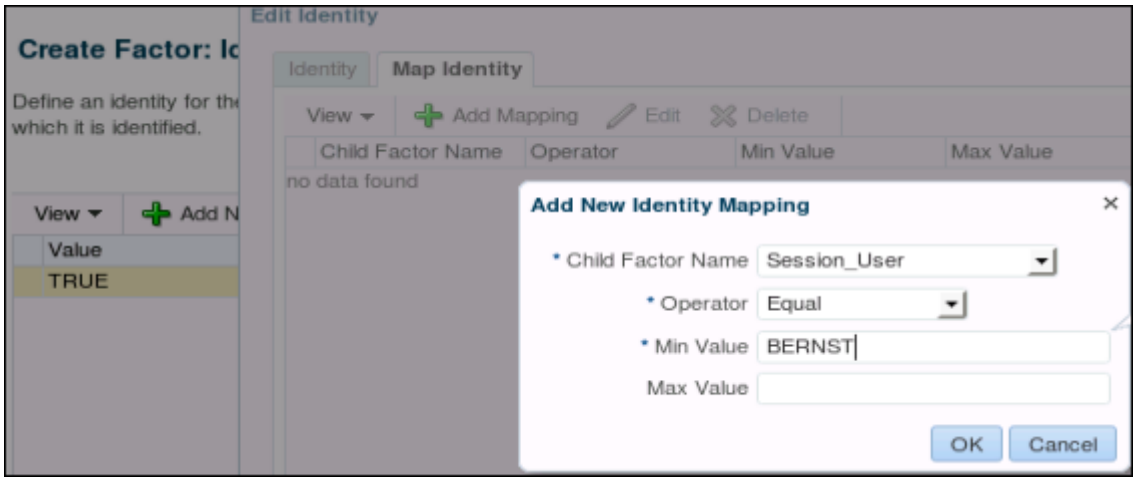
   Note: Selected screenshots are included below for additional reference.

Step	Screen/Page Description	Choices or Values
a.	Create Factor: General	Name: **IS_HR_DBA** Description: **Is the session user a designated HR_DBA (currently only BERNST is).** Factor Type: **Application** Click **Next**.
b.	Create Factor: Configurations	Factor Identification: **By Factors** Evaluation: **For Session** Factor Labeling: **By Self** Retrieval Method: Leave it blank. Click **Next**.
c.	Create Factor: Options	Click **Next**.
d.	Create Factor: Identities	Click **Add New Identity**.
e.	Add New Identity	In Identity tab: Value: **TRUE** Trust Level: **Very Trusted** Click **OK**.
f.	Create Factor: Identities	Select the **TRUE** identity and click **Edit**.
g.	Map Identity	Click **Add Mapping**.
h.	Add New Identity Mapping	Child Factor Name: **Session_User** Operator: **Equal** Min Value: **BERNST** Click **OK**.
i.	Edit Identity: Map Identity	Click **OK**.
j.	Create Factor: Identities	Click **Done**.
k.	Create Factor: Review	Review the code and click **Finish**.

e.

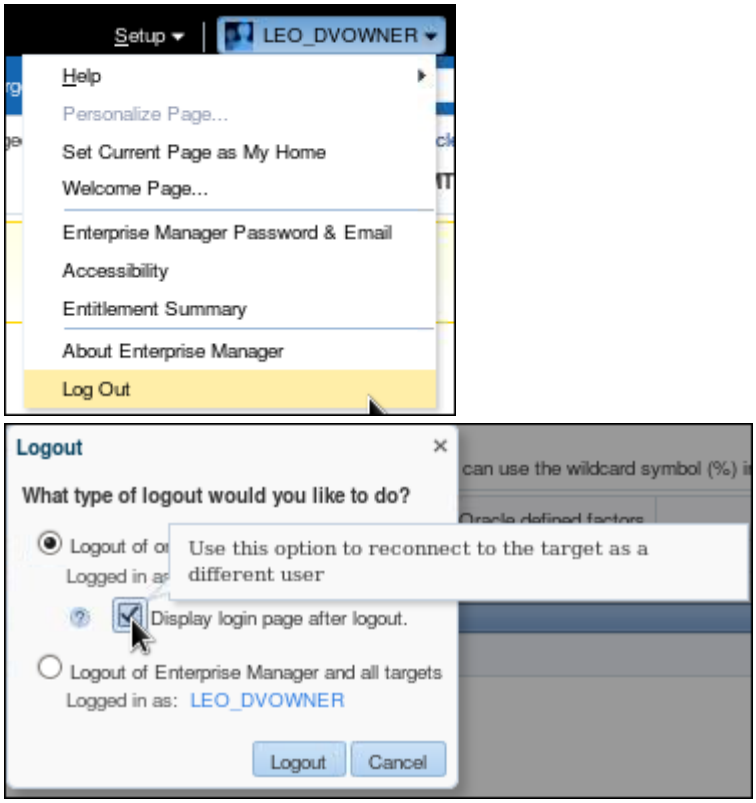


h.



k.

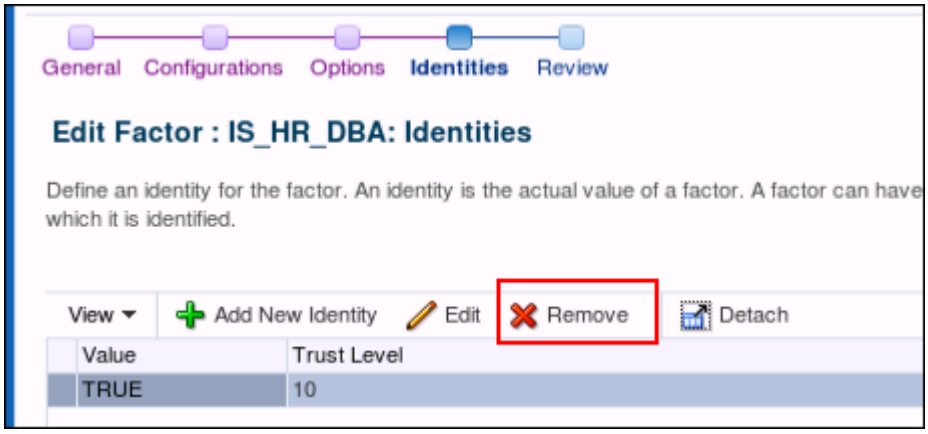
2. You receive the Confirmation message that the factor is created successfully, but you still **must** commit this operation. Log out from the `leo_dvowner` session and reconnect as `leo_dvowner`.



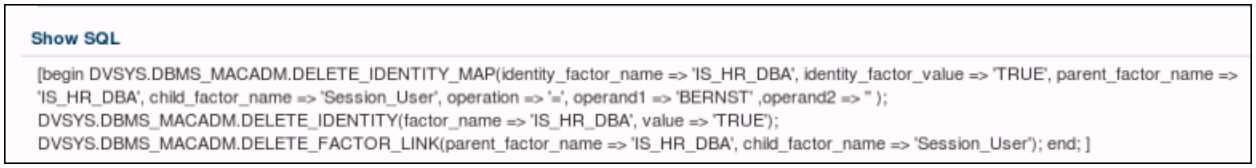
3. Test the **`IS_HR_DBA`** factor by entering the following in a SQL*Plus session:

```
SQL> connect hr
Enter password:
Connected.
SQL> select dvf.f$is_hr_dba from dual;

F$IS_HR_DBA
-----------------------------------------------------------------

SQL> connect bernst
Enter password:
Connected.
SQL> select dvf.f$is_hr_dba from dual;

F$IS_HR_DBA
-----------------------------------------------------------------
TRUE

SQL>
```

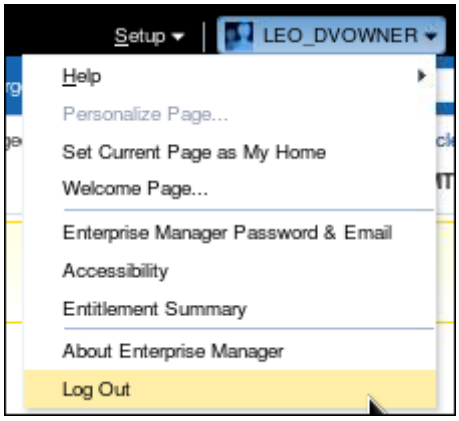
4. Delete the **IS_HR_DBA** factor, because this factor is not required in the remaining practices.
   a. Navigate to the Factors page, edit the **IS_HR_DBA** factor.
   b. On the Edit Factor: IS_HR_DBA: Identities page, select the **IS_HR_DBA** factor and then click **Remove**.

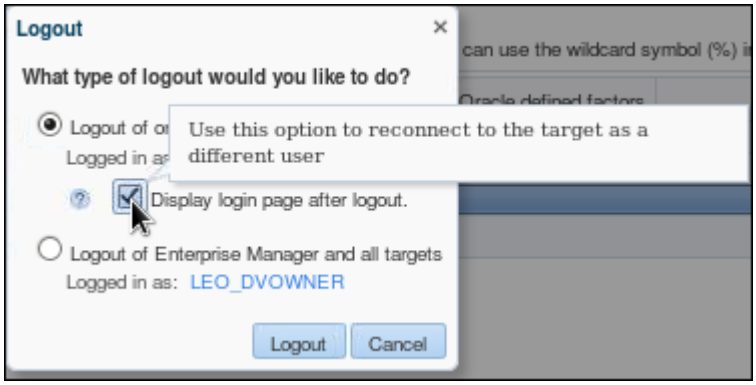


   c. Click **Done**.
   d. Review the code and then click **Finish**.



5. You receive the Confirmation message that the factor is edited successfully, but you still **must** commit this operation. Log out from the `leo_dvowner` session and reconnect as `leo_dvowner`.





Practice 8-6: Creating Time-Based Factors

In this practice, you create two factors based on time. The first one is a factor called `DayOfWeek` that returns the name of the day by using the `Time` factor type. The second factor is called `HourOfDay` and returns the current hour in 24-hour format.

1. Create a factor called **`DayOfWeek`**. Choose the `Time` factor type and a retrieval method that returns the name of the day of the week based on **`sysdate`**. Because time-based factors may need to be evaluated more often than once a session, set the evaluation to **`By Access`**.

   Navigate to the Factors page and click **Create**. Use the information in the following table to assist you.

   Hint: Use the `TO_CHAR(sysdate,'DAY')` function.

Screen/Page Description	Choices or Values
Create Factor: General	Name: **`DayOfWeek`** Description: **`Return the name of the day of the week based on sysdate.`** Factor Type: **`Time`** Click **Next**.
Create Factor: Configurations	Factor Identification: **By Method** Evaluation: **By Access** Factor Labeling: **By Self** Retrieval Method: **`TO_CHAR(sysdate,'DAY')`** Click **Done**.
Create Factor: Review	Review the code and click **Finish**.

```
[begin DVSYS.DBMS_MACADM.CREATE_FACTOR(factor_name =>
'DayOfWeek', factor_type_name => 'Time', description => 'Return
the name of the day of the week based on sysdate.',
rule_set_name => '', get_expr => 'TO_CHAR(sysdate,'DAY')',
validate_expr => '' ,identify_by => 1, labeled_by => 0,
eval_options => 1, audit_options => 1, fail_options => 1 ); end;
]
```

2. Verify that the **`DayOfWeek`** factor is set correctly. As any user, execute **`SELECT DVF.F$DayOfWeek FROM DUAL`**.

```
SQL> select DVF.F$DAYOFWEEK from dual;

F$DAYOFWEEK
--------------------------------------------------------------
THURSDAY

SQL>
```

3. Create your second time-based factor called **HourOfDay** that returns the current hour in the 24-hour format. The factor type should be `Time` and the evaluation should be `By Access`. Use the information in the following table to assist you.

   Hint: Use the `TO_CHAR(sysdate,'HH24')` function.

Screen/Page Description	Choices or Values
Create Factor: General	Name: **HourOfDay** Description: **Return the hour in 24-hour format.** Factor Type: **Time** Click **Next**.
Create Factor: Configurations	Factor Identification: **By Method** Evaluation: **By Access** Factor Labeling: **By Self** Retrieval Method: **TO_CHAR(sysdate,'HH24')** Click **Done**.
Create Factor: Review	Review the code and click **Finish**.

```
[begin DVSYS.DBMS_MACADM.CREATE_FACTOR(factor_name =>
'HourOfDay', factor_type_name => 'Time', description => 'Return
the hour in 24-hour format.', rule_set_name => '', get_expr =>
'TO_CHAR(sysdate,'HH24')', validate_expr => '' ,identify_by =>
1, labeled_by => 0, eval_options => 1, audit_options => 1,
fail_options => 1 ); end; ]
```

4. Verify that the **HourOfDay** factor is set correctly. As any user, execute **SELECT DVF.F$HourOfDay FROM DUAL**.

```
SQL> select DVF.F$HOUROFDAY from dual;

F$HOUROFDAY
------------------------------------------------------------
15

SQL> EXIT
$
```

Practices for Lesson 9: Configuring Secure Application Roles

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

You will now use secure application roles to control the users that can execute a procedure. You create a secure application role that uses a rule set, which you defined previously. You grant the ability to execute a procedure to the secure application role, which means that the session must meet the conditions of the role's associated rule set in order to execute the procedure.

Assumptions

Practices 3-1, 3-2, 3-4, **and 8-4** were successfully completed:

- Database Vault is configured.
- Test users are created.
- The `leo_dvowner` user is configured to have access to Database Vault via Cloud Control.

Practice 9-1: Managing Secure Application Roles

In this practice, you give **SMAVRIS** the ability to select from the **HR.EMPLOYEES** table but not to directly update the table. Part of **SMAVRIS**'s responsibilities includes adjusting the salaries of employees, but you do not want to allow **SMAVRIS** to perform any update on **HR.EMPLOYEES**. You create a secure application role that provides **SMAVRIS** with the ability to give employees a raise, but only by using the **HR.GIVE_RAISE** procedure.

1. As the **HR** user, issue a grant statement so that **SMAVRIS** is able to select rows from the **HR.EMPLOYEES** table.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1 is
/u01/app/oracle
$ sqlplus hr
```

```
Enter password:
SQL> GRANT SELECT ON hr.employees TO smavris;

Grant succeeded.
SQL>
```

2. As the **SMAVRIS** user, view the **HR.EMPLOYEES** table. In particular, look at the salary for the employee with **employee_id = 106**. Note the salary for this employee.

```
SQL> connect SMAVRIS
Enter password:
Connected.
SQL> SELECT last_name, salary
     FROM   hr.employees
     WHERE  employee_id = 106;

LAST_NAME                      SALARY
------------------------- ----------
Pataballa                        4800
SQL>
```

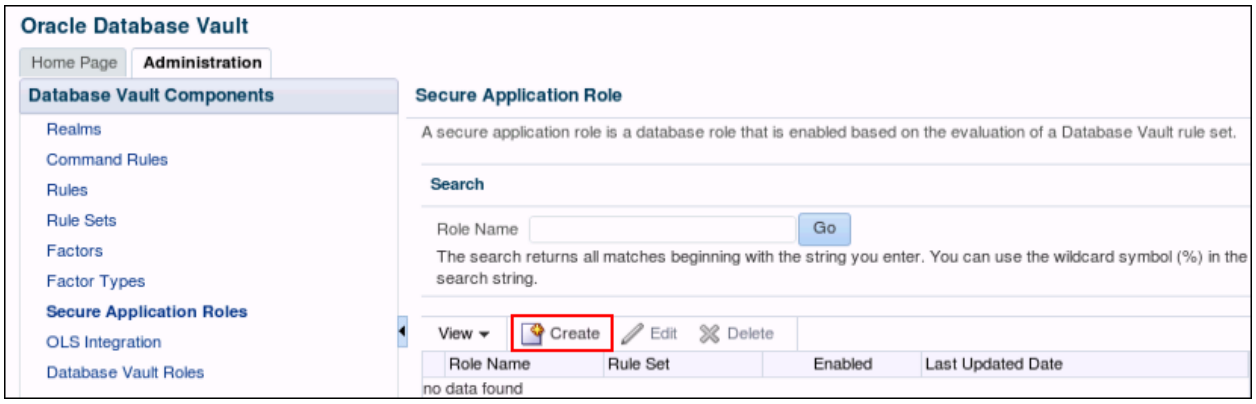
3. As the **SMAVRIS** user in SQL*Plus, use the **UPDATE** command to attempt to give every employee a 5% raise.

```
SQL> UPDATE HR.EMPLOYEES SET salary = salary * 1.05;
UPDATE HR.EMPLOYEES SET salary = salary * 1.05
          *
ERROR at line 1:
ORA-01031: insufficient privileges
SQL> EXIT
$
```

4. Still logged in to Cloud Control as the `leo_dvowner` user, navigate to Administration (tabbed page) > Secure Application Roles. Create a secure application role called **`HR_APP`** that uses the **`HR_REP_hrmaint`** rule set. Use the information in the following table to create the secure application role.

Step	Screen/Page Description	Choices or Values
a.	Secure Application Roles page	Click **Create**.
b.	Create Secure Application Role	Role: **`HR_APP`** Status: **Enabled** Rule Set: **`HR_REP_hrmaint`** Optionally, click **Show SQL**.
c.	Information	Review the code and click **OK**.
d.	Create Secure Application Role	Click **OK**. - You should receive a success message.

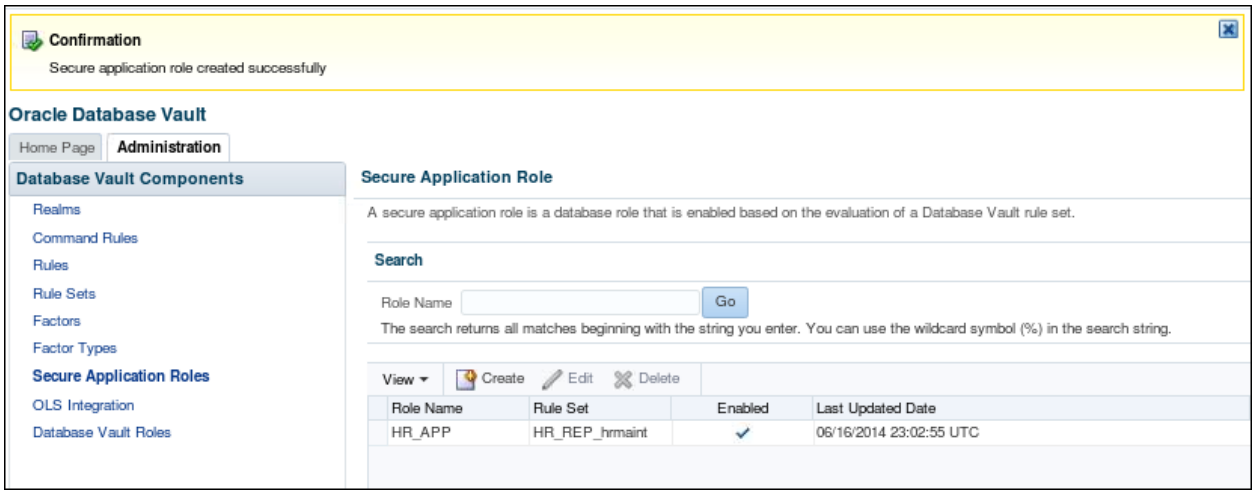
a.



b.

c.

d.



5. **SMAVRIS** needs to be able to give the employees a raise, but you do not want to give direct **UPDATE** access. Create the **HR.GIVE_RAISE** procedure while logged in as the **HR** user.

```
$ sqlplus hr
Enter password:
SQL> CREATE OR REPLACE PROCEDURE HR.GIVE_RAISE(
  Emp_id IN NUMBER,
  raise_percent IN NUMBER)
AS
  v_salary  Number(8,2);
  v_raise_pct NUMBER(6,5);
BEGIN
  v_raise_pct := raise_percent/100;
  SELECT salary INTO v_salary FROM HR.EMPLOYEES
  WHERE employee_id = EMP_ID;
  v_salary := v_salary * (1 + v_raise_pct);
  UPDATE HR.EMPLOYEES
    SET salary = v_salary
    WHERE employee_id = EMP_ID;
  EXCEPTION
    WHEN NO_DATA_FOUND THEN NULL;
END;
/
  2    3    4    5    6      2    3    4    5    6    7    8    9
10   11   12   13   14   15   16   17   18
Procedure created.

SQL>
```

6. While still connected as **HR**, grant the **EXECUTE** privilege on the **GIVE_RAISE** procedure to the **HR_APP** secure application role.

```
SQL> GRANT EXECUTE ON give_raise TO hr_app;

Grant succeeded.

SQL>
```

7. As the **SMAVRIS** user, attempt to run the **GIVE_RAISE** procedure. Attempt to give the employee with ID 106 a raise of 5%. What happens, and why?

   Note: The **GIVE_RAISE** procedure expects two parameters: the employee ID and the raise percent value.

```
SQL> connect SMAVRIS
Enter password:
Connected.
SQL> exec HR.GIVE_RAISE(106,5)
BEGIN HR.GIVE_RAISE(106,5); END;

      *
ERROR at line 1:
ORA-06550: line 1, column 7:
PLS-00201: identifier 'HR.GIVE_RAISE' must be declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored

SQL>
```

   This fails because **SMAVRIS** does not have **EXECUTE** privileges on the **HR.GIVE_RAISE** procedure.

8. To meet the requirements of the rule set on the **HR_APP** role, the session calling the **SET_ROLE** function must be known to the database as **hrmaint**. As the **SMAVRIS** user, call **DBMS_APPLICATION_INFO.SET_MODULE** to set the module to **hrmaint**. Confirm that the procedure sets the application appropriately by selecting the **sys_context('userenv','module')** from dual.

```
SQL> exec dbms_application_info.set_module('hrmaint',null)

PL/SQL procedure successfully completed.

SQL> select sys_context('userenv','module') from dual;

SYS_CONTEXT('USERENV','MODULE')
------------------------------
hrmaint

SQL>
```

9. As the **SMAVRIS** user, call **DVSYS.DBMS_MACSEC_ROLES.SET_ROLE** to enable the **HR_APP** secure application role.

```
SQL> exec dvsys.dbms_macsec_roles.set_role('HR_APP')

PL/SQL procedure successfully completed.

SQL>
```

10. Reattempt the **GIVE_RAISE** procedure, still logged in as **SMAVRIS**. What happens, and why?

```
SQL> EXEC HR.GIVE_RAISE(106, 5)

PL/SQL procedure successfully completed.

SQL>
```

11. Verify that the raise was applied. Query for the salary of employee number 106 and compare it with the original salary that you noted from an earlier query.

```
SQL> SELECT last_name, salary FROM HR.EMPLOYEES
     WHERE EMPLOYEE_ID = 106;
  2
LAST_NAME                      SALARY
------------------------- ----------
Pataballa                        5040

SQL>
```

Note that previously this salary was 4800. Now it is 5040, which is 5% higher. Therefore, the 5% raise has taken effect.

12. Roll back your changes and exit the SQL*Plus session.

```
SQL> rollback;

Rollback complete.

SQL> exit
$
```

Practices for Lesson 10: Auditing with Database Vault Reports

Chapter 10

Practices for Lesson 10: Overview

Practices Overview

These practices will guide you through the several reports that are available in Database Vault. To provide some data to be viewed in these reports, you run scripts to create some Database Vault components that are not correctly configured and users operations that violate Database Vault restrictions. You can run others that are not specified here, if you want to.

Assumptions

Practices 3-1, 3-2, and 3-4 were successfully completed:

- Database Vault is configured.
- Test users are created.
- The `leo_dvowner` user is configured to have access to Database Vault via Cloud Control

Practice 10-1: Viewing Configuration Issues Reports

In this practice, you view the various reports that are available in Oracle Database Vault from EM Cloud Control. You create a new user to act as an auditor who requires the ability to view the various Database Vault reports. Then you run a script to create some incorrectly configured Database Vault components so as to create more data for the reports.

Mostly, you should see results similar to the screenshots shown in this practice; however, depending on any additional Database Vault activities that you have performed, the results may differ.

1. With the separation of duties, three professionals work together to set up this user:
   - As the Database Vault account manager user (**bea_dvacctmgr**) creates a new user named **SECVIEW** with the password **oracle_4U**. This user serves the purpose of a security auditor who views the enforcement audit reports and configuration issues reports but does not have the privileges to change the Database Vault configuration.
   - Only the Database Vault account owner (**leo_dvowner)** can grant the DV_SECANALYST role.
   - Then the DBA_PSMITH user grants system privileges.

   Verify that this user can view the reports and the edit pages for the various Database Vault objects, but cannot change the Database Vault objects.

   a. Create the SECVIEW user.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus bea_dvacctmgr
Enter password:
SQL> CREATE USER secview IDENTIFIED BY oracle_4U;

User created.
SQL>
```

   b. Only the user with the DV_OWNER role can grant or revoke the DV_SECANALYST role to or from another user. Connect as leo_dvowner to grant the DV_SECANALYST role to the user.

```
SQL> connect leo_dvowner
Enter password:
Connected.
SQL> GRANT DV_SECANALYST TO secview;

Grant succeeded.
SQL>
```

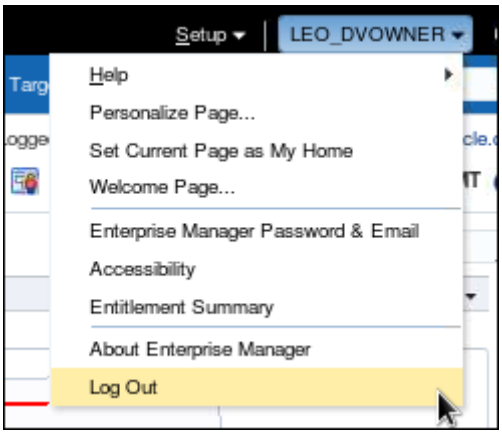
   c. As the DBA_PSMITH user, grant the CREATE SESSION and SELECT ANY DICTIONARY system privileges.

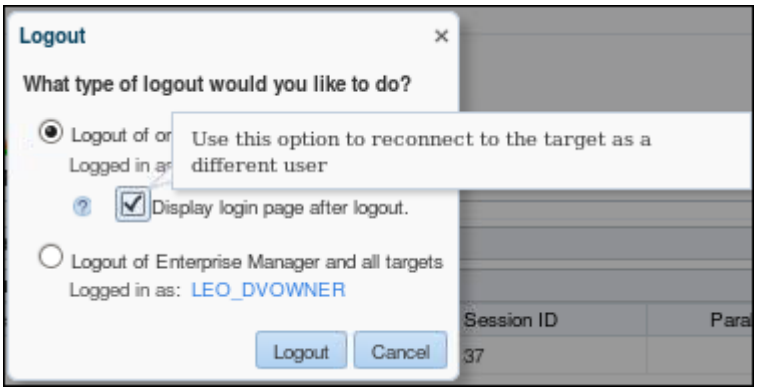
```
SQL> conn dba_psmith
Password:
Connected.
```

```
SQL> GRANT create session, SELECT ANY DICTIONARY  TO secview;

Grant succeeded.
SQL>
```

d. If you are still logged in to EM Cloud Control as `leo_dvowner`, log out and then log in as the `secview` user.

1) On the top-right, under `leo_dvowner`, click **Log Out**. Then select **Logout of orcl** and **Display login page after logout**. Click **Logout**.





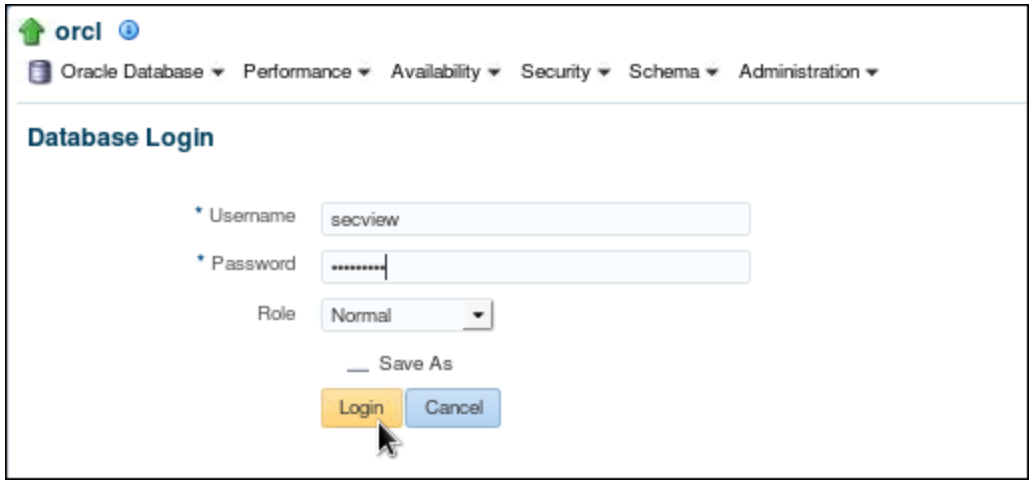
2) On the Login page, enter the following:

Username: **`secview`**

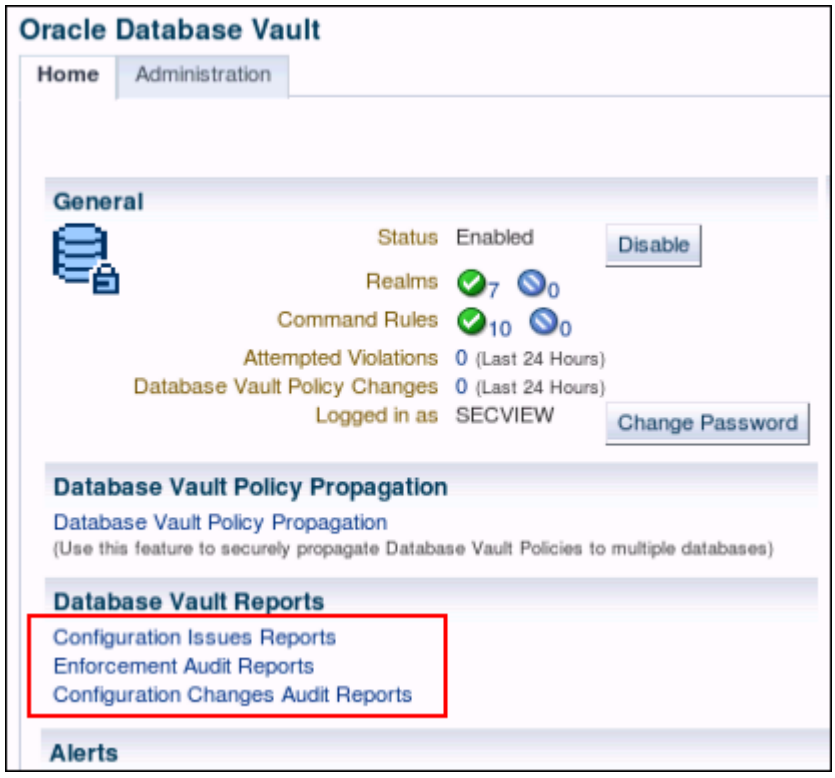
Password: **`oracle_4U`**

Role: Normal.

Click **Login**.



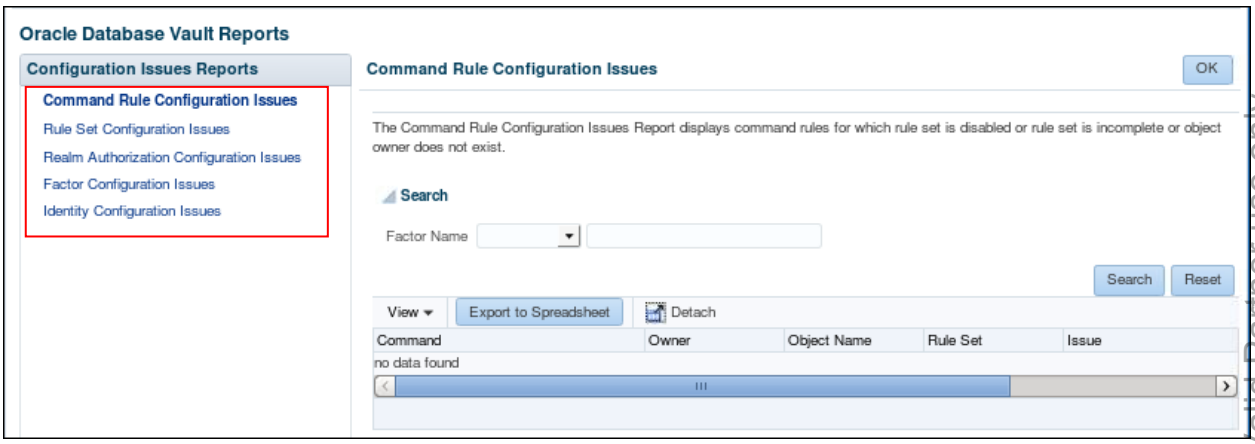
e. Navigate to Security > Database Vault.



f. On the Oracle Database Vault Home page in the Database Vault Reports section, you see three types of reports:
   - Configuration Issues Reports
   - Enforcement Audit Reports
   - Configuration Changes Audit Reports

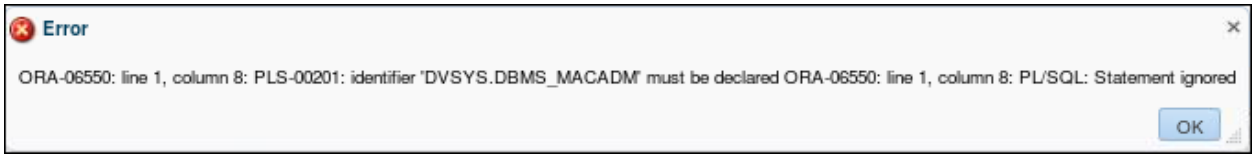
g. Click the **Configuration Issues Reports** link. You see five types of reports:
   - Command Rule Configuration Issues

- Rule Set Configuration Issues
- Realm Authorization Configuration Issues
- Factor Configuration Issues
- Identity Configuration Issues



h. Click **OK** to go back to the Home page. Navigate to the Administration tabbed page to attempt to change one of the factors. Note that the Administration link is available. The `secview` user can view information.

i. But when you attempt to edit the definition of a factor (for example, attempt to change the Factor Type and click **Done** and **Finish**), then you get an error message.



2. There are several ways to view a script before executing it. Logged in to SQL*Plus, you can use the `!` (exclamation mark) before the command. (If you are not in the `labs` directory, you must use the path and file name.)

```
SQL> !cat $HOME/labs/lab_10_01_objects.sql
-- DISCLAIMER:
-- This script is provided for educational purposes only. It is
-- NOT supported by Oracle World Wide Technical Support.
-- The script has been tested and appears to work as intended.
-- You should always run new scripts on a test instance
initially.
-- Create misconfigured DV objects

CONNECT leo_dvowner/oracle_4U

EXECUTE DVSYS.DBMS_MACADM.CREATE_REALM('OE_REALM',-
'Protect the OE schema', 'Y', 1);

EXECUTE DVSYS.DBMS_MACADM.CREATE_REALM('SH_REALM',-
'Protect the SH schema', 'Y', 1);
```

```
EXECUTE DVSYS.DBMS_MACADM.ADD_OBJECT_TO_REALM('OE_REALM',-
 'OE', 'CUSTOMERS', 'TABLE');

EXECUTE DVSYS.DBMS_MACADM.ADD_OBJECT_TO_REALM('SH_REALM',-
 'SH', '%', '%');

EXECUTE DVSYS.DBMS_MACADM.CREATE_RULE_SET('OE_AUTH_RULES',-
 'Rule set for OE schema authorization','N', 1, 1, 1, NULL,-
  NULL,NULL, NULL );

EXECUTE DVSYS.DBMS_MACADM.CREATE_RULE_SET('SH_AUTH_RULES',-
 'Rule set for SH schema authorization','N', 1, 1, 1, NULL,-
  NULL,NULL, NULL );

EXECUTE DVSYS.DBMS_MACADM.ADD_AUTH_TO_REALM('SH_REALM',-
  'WSMITH', 'SH_AUTH_RULES');

EXECUTE DVSYS.DBMS_MACADM.ADD_AUTH_TO_REALM('SH_REALM',-
  'SH', NULL, 1);

EXECUTE DVSYS.DBMS_MACADM.CREATE_RULE('SH_AUTH',-
 'DVF.F$SESSION_USER = ''SH''');

EXECUTE DVSYS.DBMS_MACADM.CREATE_RULE('SH_PROXY',-
 'DVF.F$PROXY_USER = ''SH''');

EXECUTE DVSYS.DBMS_MACADM.ADD_RULE_TO_RULE_SET('SH_AUTH_RULES',-
 'SH_AUTH');

EXECUTE DVSYS.DBMS_MACADM.DELETE_FACTOR('HourOfDay');

EXECUTE DVSYS.DBMS_MACADM.CREATE_FACTOR('HourOfDay', 'Time',-
 'Return the hour of the day on the server in 24hr format',-
  NULL, 'TO_CHAR(SYSDATE,''HH24'')', NULL, 1, 0, 1, 0, 1);

EXECUTE DVSYS.DBMS_MACADM.CREATE_FACTOR('LUNCHTIME', 'Time',-
'Determine Lunch time for multiple shifts', NULL, NULL, NULL,-
 2, 0, 0, 0, 1);

EXECUTE DVSYS.DBMS_MACADM.CREATE_IDENTITY('LUNCHTIME',-
 'FIRST_SHIFT_LUNCH', 5);
```

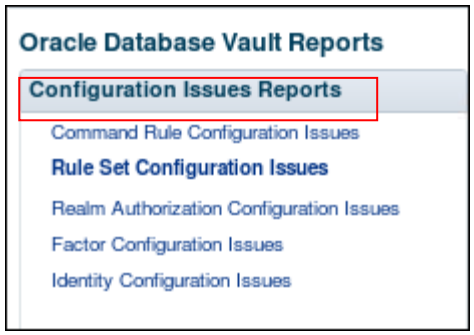
```
EXECUTE DVSYS.DBMS_MACADM.CREATE_IDENTITY('LUNCHTIME',-
 'SECOND_SHIFT_LUNCH', 5);

EXECUTE DVSYS.DBMS_MACADM.ADD_FACTOR_LINK('LUNCHTIME',
'HourOfDay',NULL);

EXECUTE DVSYS.DBMS_MACADM.CREATE_IDENTITY_MAP('LUNCHTIME',-
 'FIRST_SHIFT_LUNCH', 'LUNCHTIME', 'HourOfDay', '=', '12',
NULL);

EXECUTE DVSYS.DBMS_MACADM.CREATE_FACTOR('IS_OE_USER',
'Application',-
 'Return OE fundtional role if user is authorized OE user',-
  'OE_AUTH_RULES', 'hr.get_job_id', NULL, 1, 0, 1, 0, 1);

EXECUTE DVSYS.DBMS_MACADM.CREATE_RULE_SET('IS_SH_USER',-
'Ensures the current user is a valid user of SH schema',-
'Y',2,0,1,NULL,NULL,0,NULL);

EXECUTE DVSYS.DBMS_MACADM.CREATE_COMMAND_RULE('SELECT',-
'IS_SH_USER','SH','%','Y')

EXECUTE
DVSYS.DBMS_MACADM.CREATE_ROLE('SH_APP_ROLE','Y','SH_AUTH_RULES')
;

COMMIT;

SQL>
```

3. Execute the **lab_10_01_objects.sql** script to create some Database Vault components that are not correctly configured.

```
SQL> @$HOME/labs/lab_10_01_objects.sql
Connected.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
```

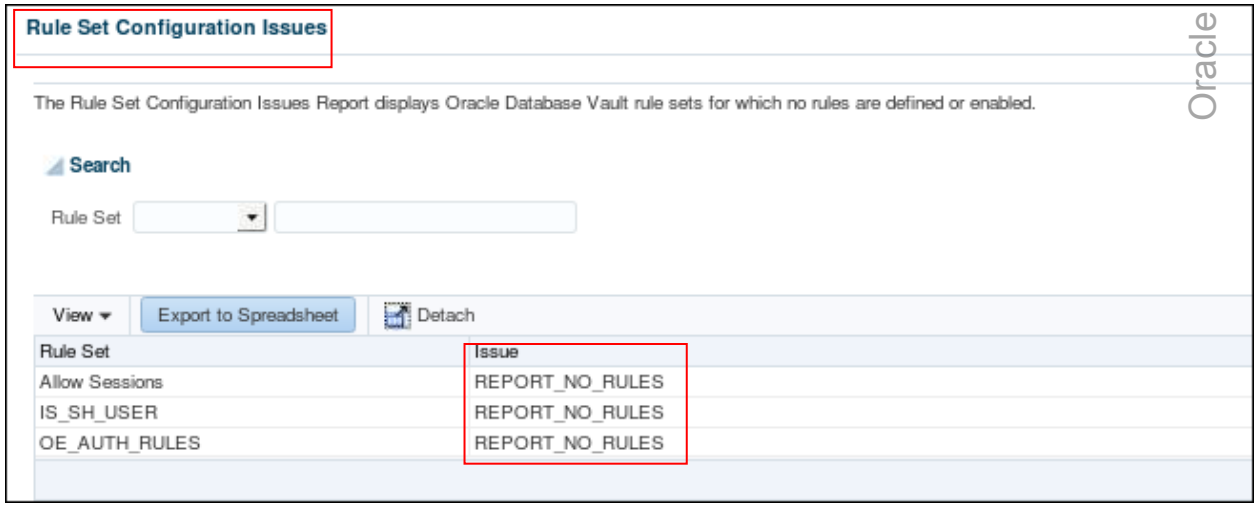
```
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
Commit complete.
SQL>
```

4. In the Oracle Database Vault Reports section, click each of the reports listed in **Configuration Issues Reports**.

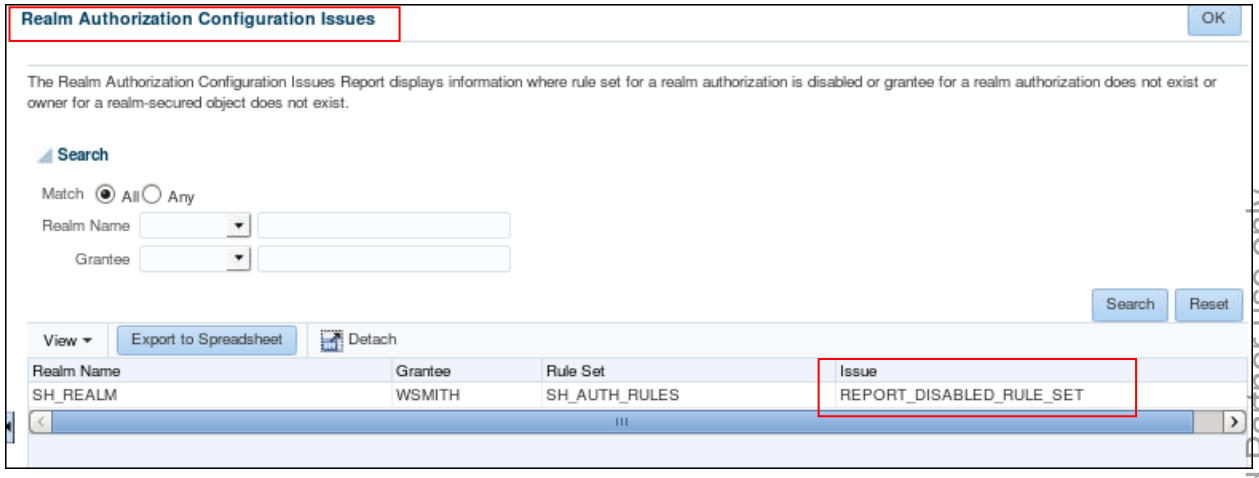


   a. Click the **Command Rule Configuration Issues** link. There is no command rule configuration issue.

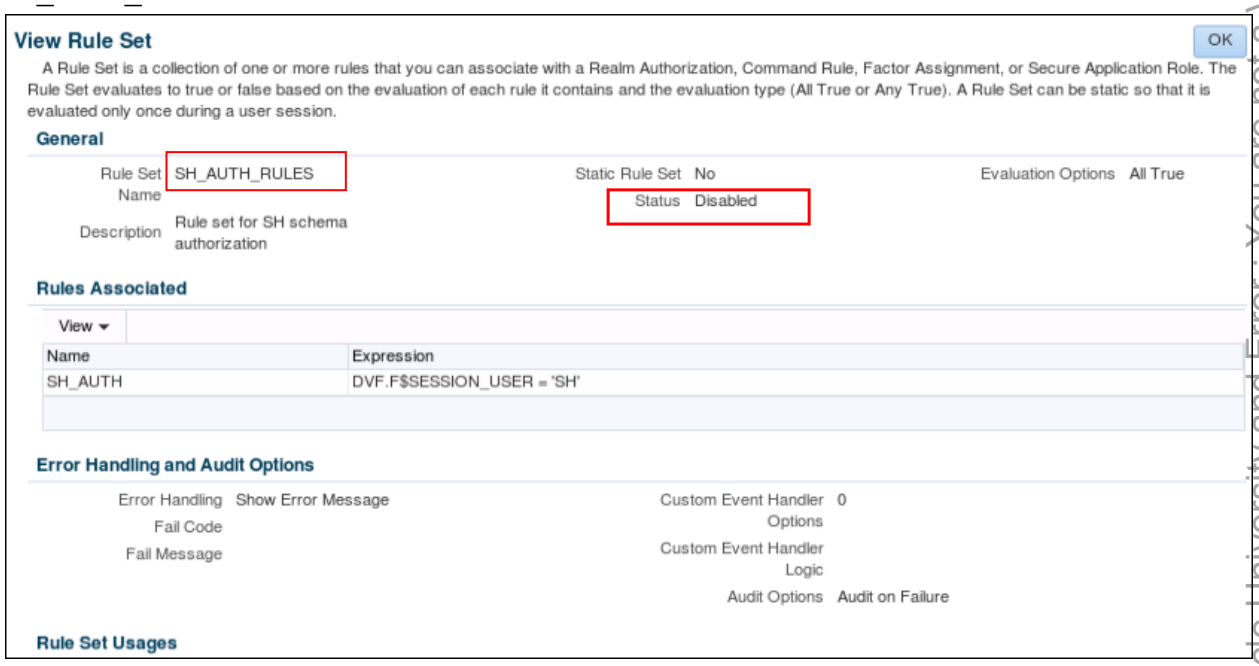
   b. Click the **Rule Set Configuration Issues** link. The Rule Set Configuration Issues report displays Oracle Database Vault rule sets for which no rules are defined or enabled. You can investigate in the Administration tab by reviewing each of the reported rule sets. You find that `IS_SH_USER` and `OE_AUTH_RULES` have no rule.



c. Click the **Realm Authorization Configuration Issues** link. It informs that the `SH_REALM` realm authorization has a rule set that is disabled.

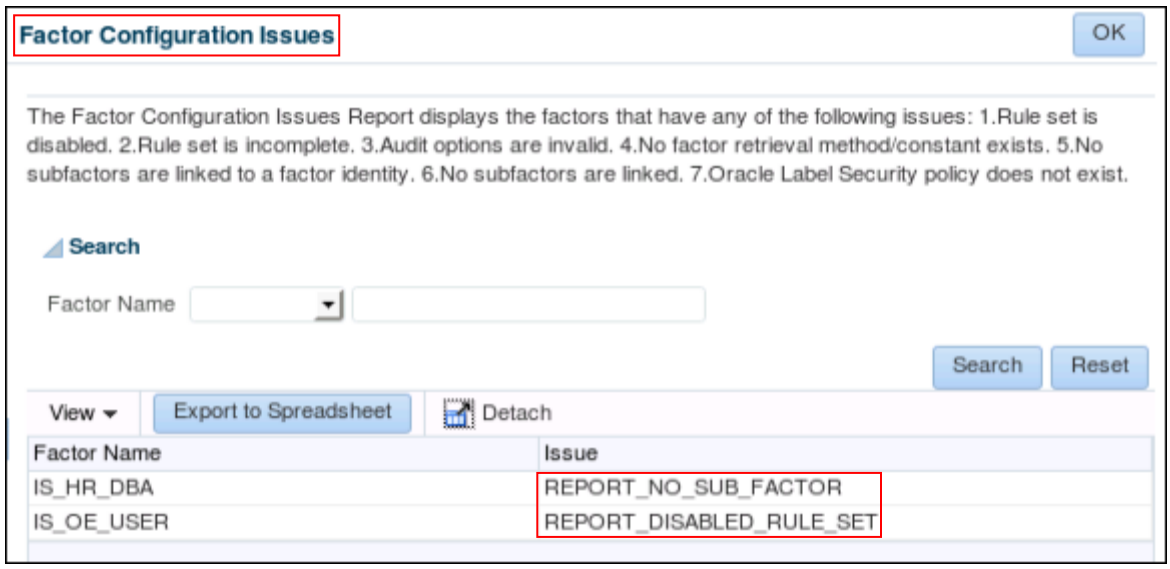


You can investigate in the Administration tab by reviewing the reported `SH_AUTH_RULES` rule set.

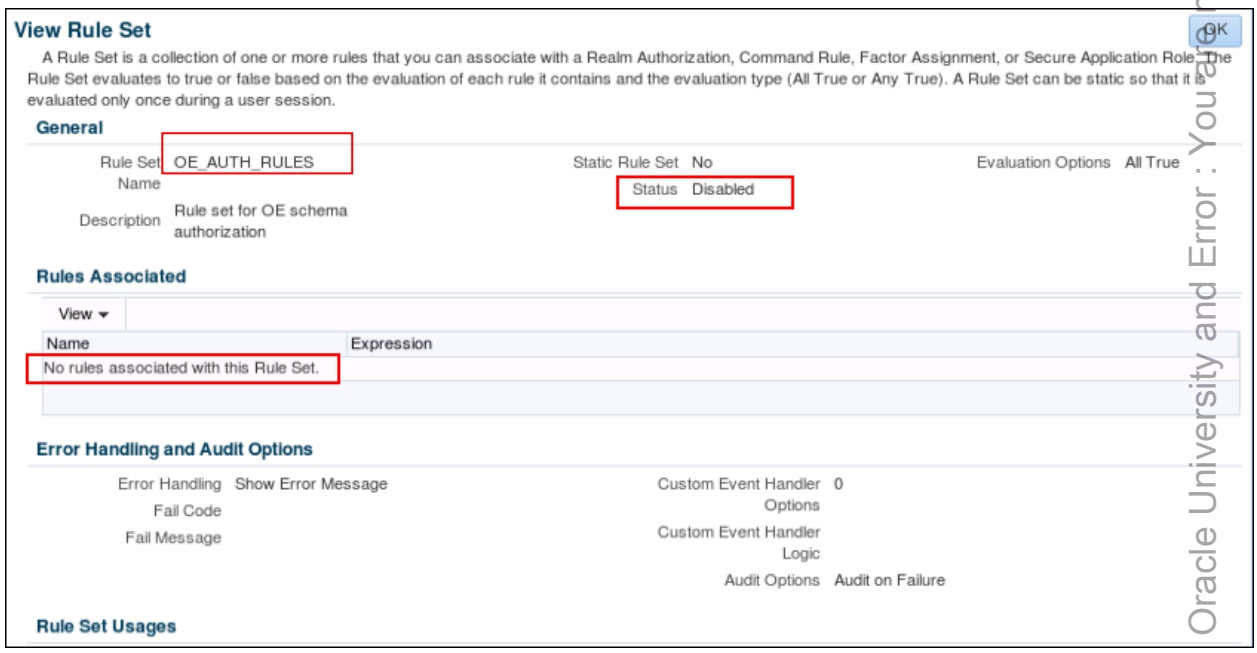


d. Click the **Factor Configuration Issues** link. The Factor Configuration Issues report displays the factors that have any of the following issues:

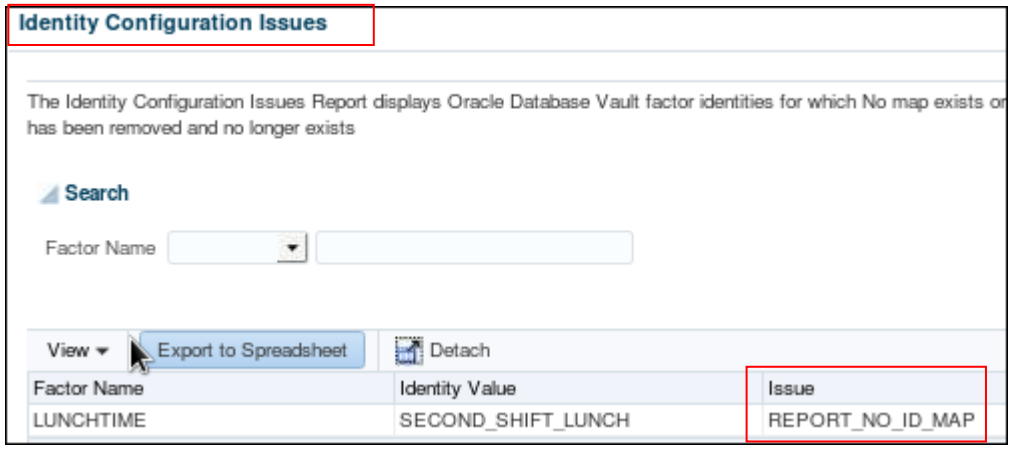
- Rule set is disabled.
- Rule set is incomplete.
- Audit options are invalid.
- No factor retrieval method/constant exists.
- No subfactors are linked to a factor identity.
- No subfactors are linked.
- Oracle Label Security policy does not exist.



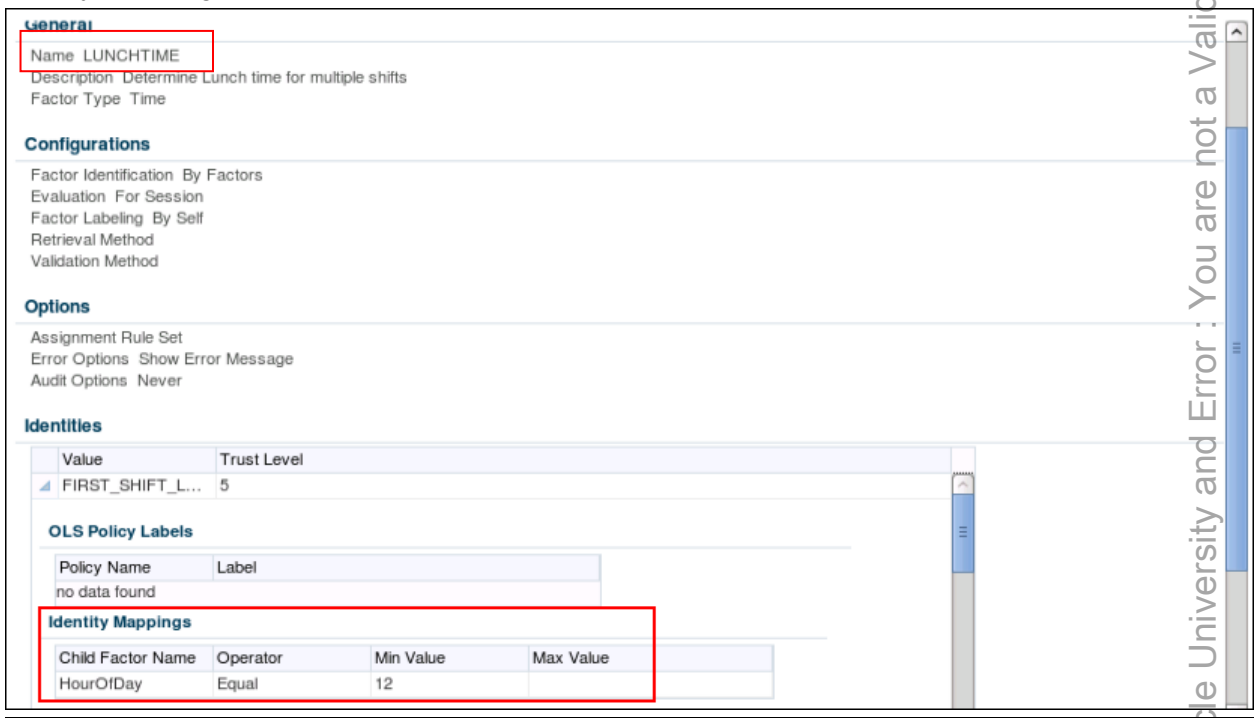
You can investigate in the Administration tab by reviewing the reported `OE_AUTH_RULES` rule set. In this case, the `IS_OE_USER` factor is using the `OE_AUTH_RULES` assignment rule set, which has no rule assigned and is therefore incomplete and, moreover, disabled.



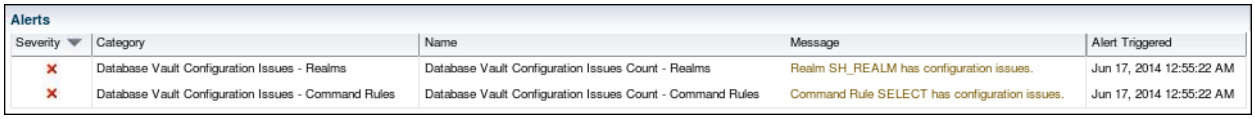
e. Click the **Identity Configuration Issues** link. The Identity Configuration Issues report displays Oracle Database Vault factor identities for which no map exists or for which Label identity for the Oracle Label Security label has been removed and no longer exists.



You can investigate in the Administration tab by reviewing the reported `LUNCHTIME` factor. In this case, the `SECOND_SHIFT_LUNCH` identity for the `LUNCHTIME` factor has no map, whereas the `FIRST_SHIFT_LUNCH` has one identity mapping.



5. After some time, alerts are reported in the Alerts section below the Database Vault Reports. You may see the alerts in the Home page.



Alerts

Severity	Category	Name	Message	Alert Triggered
✕	Database Vault Configuration Issues - Realms	Database Vault Configuration Issues Count - Realms	Realm SH_REALM has configuration issues.	Jun 17, 2014 12:55:22 AM
✕	Database Vault Configuration Issues - Command Rules	Database Vault Configuration Issues Count - Command Rules	Command Rule SELECT has configuration issues.	Jun 17, 2014 12:55:22 AM

Practice 10-2: Viewing Enforcement Audit Reports

Overview

In this practice, you will view various enforcement audit reports.

You first run a script that executes, as various users, SQL commands that violate configured Database Vault objects so as to create more data for the enforcement audit reports. Mostly, you should see results similar to the screenshots shown in this practice; however, depending on any additional Database Vault activities that you have performed, the results may differ.

Assumptions

The `secview` user has been created in Practice 10-1.

Tasks

1. Because the databases are working in a unified auditing environment, for Database Vault to report on Database Vault violations, unified audit policies need to be created and enabled. Create and enable unified audit policies for Database Vault objects—like realms, rule sets and factors—by executing the **`lab_10_02_01.sql`** script.

```
SQL> !cat /home/oracle/labs/lab_10_02_01.sql
-- Create audit policies on DV objects like realms, RS, factors

Connect / as sysdba
CREATE AUDIT POLICY audpol_realms ACTIONS
COMPONENT = dv realm violation on "HR_Schema",
               realm violation on "SH_REALM",
               realm violation on "OE_REALM";
AUDIT POLICY audpol_realms;

CREATE AUDIT POLICY audpol_rulesets ACTIONS
COMPONENT = dv rule set failure on "SH_AUTH_RULES",
               rule set failure on "OE_AUTH_RULES",
               rule set failure on "Non_Work_Hours",
               rule set failure on "HAS_APP_ROLE",
                  rule set failure on "IS_SH_USER",
               rule set failure on "HR_REP_hrmaint"   ;
AUDIT POLICY audpol_rulesets;

CREATE AUDIT POLICY audpol_factors ACTIONS
COMPONENT = dv factor error on "HourOfDay",
               factor error on "DayOfWeek",
               factor error on "LUNCHTIME",
                  factor error on "IS_OE_USER",
               factor error on "JOBROLE" ;
AUDIT POLICY audpol_factors;
```

```
CREATE AUDIT POLICY audpol_predefined_realms ACTIONS
COMPONENT = dv realm violation on "Oracle Database Vault",
               realm violation on "Oracle Enterprise Manager",
               realm violation on "Database Vault Account
Management",
               realm violation on "Oracle Default Schema
Protection Realm",
                  realm violation on "Oracle System Privilege
and Role Management Realm",
                  realm violation on "Oracle Default Component
Protection Realm";
AUDIT POLICY audpol_predefined_realms;

SQL>
```

```
SQL> @/home/oracle/labs/lab_10_02_01.sql
Connected.
Audit policy created.
Audit succeeded.
Audit policy created.
Audit succeeded.
Audit policy created.
Audit succeeded.
Audit policy created.
Audit succeeded.
SQL>
```

2. Run the **lab_10_02_02.sql** script, which executes, as various users, SQL commands that violate configured Database Vault objects.

   Note: The SQL scripts are not displayed here, so that you have a more realistic scenario to analyze.

```
SQL> set echo on
SQL> @/home/oracle/labs/lab_10_02_02.sql
Connected.
Select * from hr.employees
                *
ERROR at line 1:
ORA-01031: insufficient privileges

Update hr.employees set salary = salary*.75
          *
ERROR at line 1:
ORA-01031: insufficient privileges
```

```
Grant succeeded.

Table created.

Connected.
select * from sh.sales
                 *
ERROR at line 1:
ORA-00942: table or view does not exist

update sh.sales set AMOUNT_SOLD = AMOUNT_SOLD*1.02
          *
ERROR at line 1:
ORA-00942: table or view does not exist

Connected.
select * from hr.employees
                 *
ERROR at line 1:
ORA-01031: insufficient privileges

Grant all on sh.sales to Kpartner
                *
ERROR at line 1:
ORA-47401: Realm violation for GRANT on SH.SALES

SQL>
```

3. Run the **lab_10_02_03.sql** script, which grants privileges that should not normally be granted and attempts to alter some system parameters. Some errors are generated, creating more audit information for the reports.

```
SQL> @/home/oracle/labs/lab_10_02_03.sql
SQL> -- Unusual grants, attempted PARAMETER CHANGES
SQL>
SQL> Connect / as sysdba
Connected.
SQL>
SQL> GRANT sysdba to bernst;

Grant succeeded.
SQL>
SQL> GRANT exempt access policy to AHUNOLD;
```

```
Grant succeeded.
SQL>
SQL> Grant DBA to KPARTNER;

Grant succeeded.
SQL>
SQL> ALTER SYSTEM SET AUDIT_SYS_OPERATIONS=FALSE SCOPE=SPFILE;
ALTER SYSTEM SET AUDIT_SYS_OPERATIONS=FALSE SCOPE=SPFILE
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
SQL> ALTER SYSTEM SET UTL_FILE_DIR=* SCOPE=SPFILE;
ALTER SYSTEM SET UTL_FILE_DIR=* SCOPE=SPFILE
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
SQL> connect HR/oracle_4U
Connected.
SQL>
SQL> GRANT ALL on hr.employees to WSMITH with grant option;

Grant succeeded.
SQL>
SQL> GRANT ALL on hr.get_job_id to WSMITH;

Grant succeeded.
SQL>
```

4. To get the list of the enforcement audit records, connect as `secview`. You will use the `SYS.DV$ENFORCEMENT_AUDIT` view, which is almost the same as the `DVSYS.DV$ENFORCEMENT_AUDIT` view except that it captures Database Vault–related audit records from the unified audit trail instead of `DVSYS.AUDIT_TRAIL$`.

```
SQL> CONNECT secview
Enter password:
Connected.
SQL> set pages 100
 col OBJ_NAME format A10
 col DV_ACTION_NAME format A24
```

```
 col "Vault Object" format A10
 col DV_RULE_SET_NAME format A14
 col "CODE " format 99999
 col USERID format A8
SQL> SQL> SQL> SQL> SQL> SQL> SQL>
SQL>
SQL> SELECT USERID, OBJ_NAME, DV_ACTION_NAME,
            DV_ACTION_OBJECT_NAME "Vault Object",
            DV_RULE_SET_NAME, DV_RETURN_CODE "CODE "
     FROM   sys.dv$enforcement_audit
     WHERE  DV_RETURN_CODE <>0;
  2    3    4    5

USERID OBJ_NAME    DV_ACTION_NAME            Vault Obje
------ ----------- ------------------------ ----------
DV_RULE_SET_NA    CODE
-------------- ------
WSMITH              Factor Assignment Audit  JOBROLE
                  47394
SYS        EMPLOYEES Realm Violation Audit   HR_Schema
             1031
SYS        EMPLOYEES Realm Violation Audit   HR_Schema
             1031
WSMITH     EMPLOYEES Realm Violation Audit   HR_Schema
             1031
WSMITH     SALES     Realm Violation Audit    SH_REALM
SH_AUTH_RULES   47401


SQL>
```

Note: You can review the earlier output and see that `1031` is the ORA-01031: insufficient privileges error.

Practice 10-3: Viewing Database Vault Configuration Changes

Overview

In this optional practice, you will detect who performed any change, deletion, or creation of any type of Database Vault objects, such as realm deletion, factor update, command rule creation, or rule update.

Assumptions

The `secview` user has been created in practice 10-1.

The scripts have been executed in practice 10-2.

Tasks

1. Connect as `secview` to report all Database Vault object changes. This information relies on the `SYS.DV$CONFIGURATION_AUDIT` view, which is almost the same as the `DVSYS.DV$CONFIGURATION_AUDIT` view except that it captures Database Vault–related audit records from the unified audit trail instead of `DVSYS.AUDIT_TRAIL$`.

```
SQL> CONNECT secview
Enter password:
Connected.
SQL> set pages 100
     col DV_ACTION_NAME format A26
     col DV_RULE_SET_NAME format A14
     col DV_ACTION_OBJECT_NAME format A40
     col USERID format A11
     col SQL_TEXT FORMAT A78
SQL> SQL> SQL> SQL> SQL>
SQL>
SQL> SELECT USERID, DV_ACTION_NAME, DV_ACTION_OBJECT_NAME,
            SQL_TEXT, DV_RULE_SET_NAME
     FROM  sys.dv$configuration_audit
     ORDER BY 1,2,3;

>>>> output removed to avoid cluttering this activity guide <<<
. . .
LEO_DVOWNER Rule Set Creation Audit    Non_Work_Hours
begin  DECLARE x VARCHAR2(40);static_option BOOLEAN := FALSE;
BEGIN x:=:1 ; IF
 x


LEO_DVOWNER Rule Set Creation Audit    OE_AUTH_RULES
BEGIN DVSYS.DBMS_MACADM.CREATE_RULE_SET('OE_AUTH_RULES',  'Rule
set for OE sch
em
```

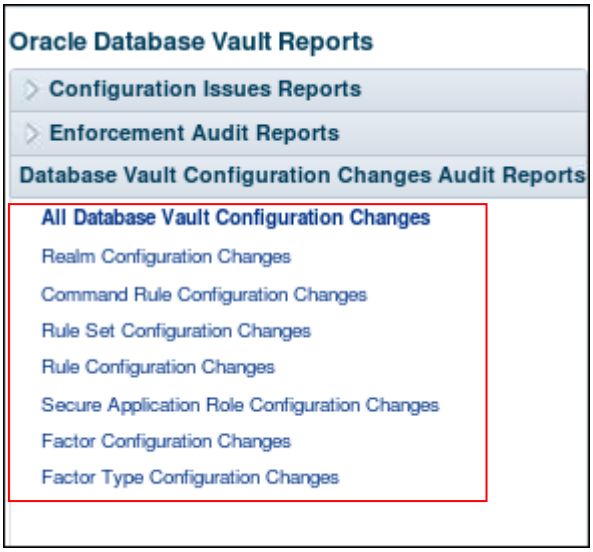
```
LEO_DVOWNER Rule Set Creation Audit    SH_AUTH_RULES
BEGIN DVSYS.DBMS_MACADM.CREATE_RULE_SET('SH_AUTH_RULES',  'Rule
set for SH sch
em

85 rows selected.
SQL>
```

Note: Your row number may be different.

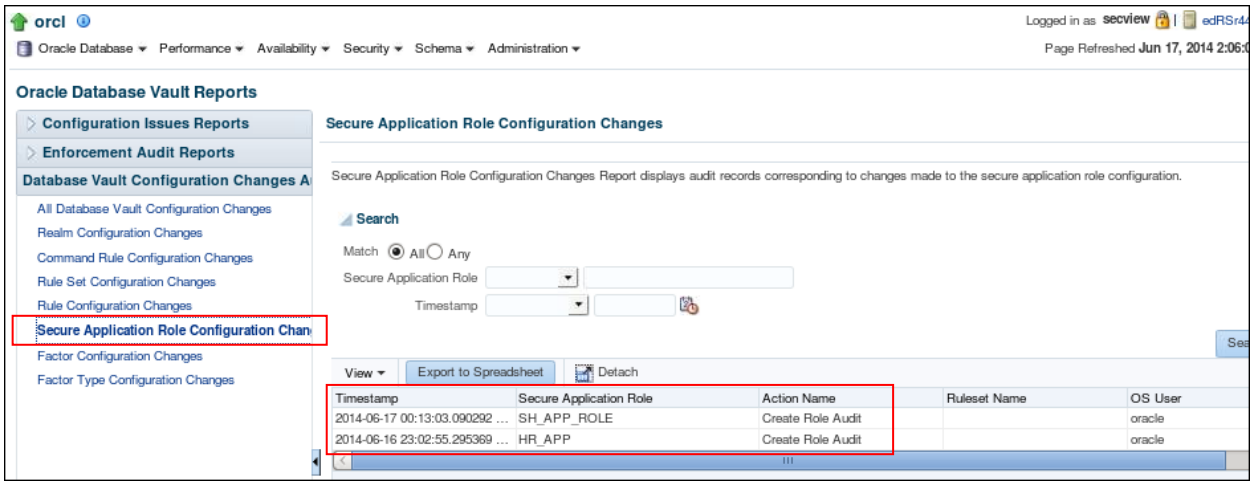
2. Focus on Secure Application Role creation, deletion, or update. Use either SQL query or EM Cloud Control Oracle Database Vault Reports.
   a. Use a SQL query.

```
SQL> SELECT USERID, DV_ACTION_NAME, DV_ACTION_OBJECT_NAME,
            SQL_TEXT, DV_RULE_SET_NAME
     FROM  sys.dv$configuration_audit
     WHERE DV_ACTION_NAME like '%Role%';
  2    3    4

USERID        DV_ACTION_NAME           DV_ACTION_OBJECT_NAME
----------- -------------------------- -------------------------
---------------
SQL_TEXT
-----------------------------------------------------------------
--------------
DV_RULE_SET_NA
--------------
LEO_DVOWNER Create Role Audit          HR_APP
begin
DVSYS.DBMS_MACADM.CREATE_ROLE(role_name=>DBMS_ASSERT.ENQUOTE_NAM
E(:1 ),
 e



LEO_DVOWNER Create Role Audit          SH_APP_ROLE
BEGIN
DVSYS.DBMS_MACADM.CREATE_ROLE('SH_APP_ROLE','Y','SH_AUTH_RULES')
; END;

SQL>
```

   b. Use EM Cloud Control. Once in Database Vault, navigate to Home > Oracle Database Vault Reports.
      1) Click **Configuration Changes Audit Reports**.



2) Click **Secure Application Role Configuration Changes**.



Practice 10-4: Viewing General Security Reports

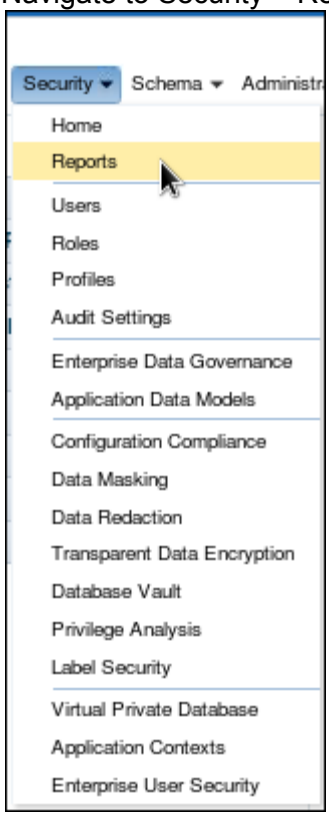
Overview

In this optional practice, you can explore some of the numerous general security reports available in the Security area.

Note: Tasks 1 - 3 are optional, but 4 is mandatory.

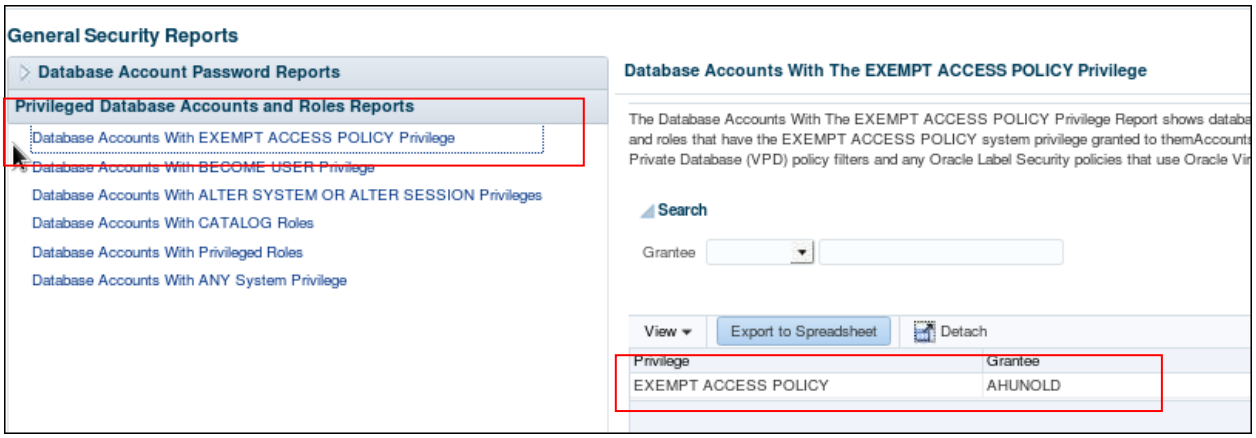
Tasks

1. There are many other General Security reports that help you identify common security problems. Many of these reports are considered standard reports needed to view the security aspects of the database.
   a. Navigate to Security > Reports.



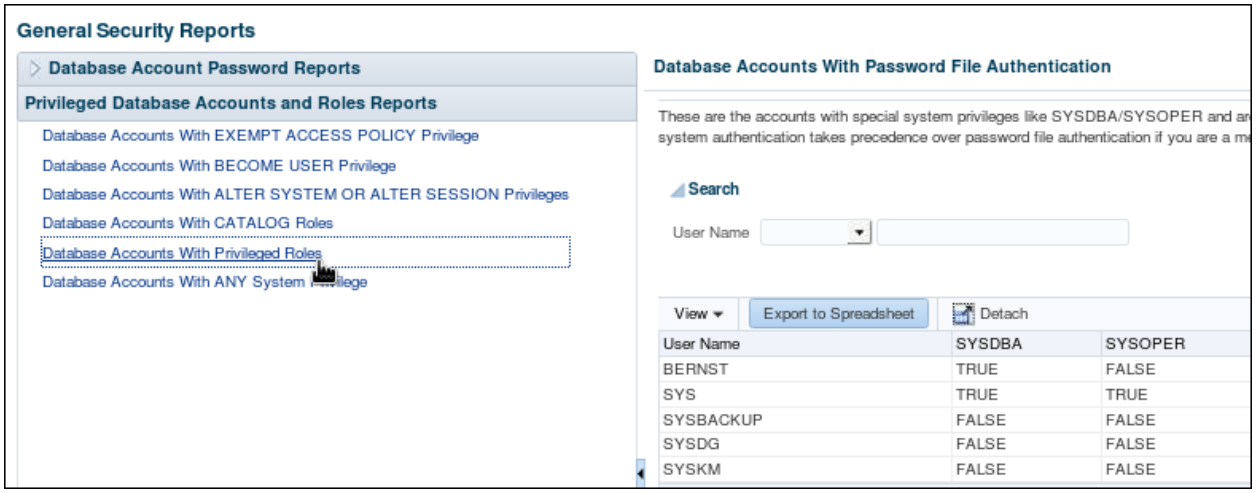
   b. Review the list of all the general security reports, expand the groups to see more reports.

2. Click **Privileged Database Accounts and Roles Reports** and then focus on the database accounts being granted the `EXEMPT ACCESS POLICY` system privilege.



Decide whether `AHUNOLD` should still be granted the `EXEMPT ACCESS POLICY` system privilege.

3. Click **Database Accounts With Privileges Roles**.



You notice that `BERNST` is granted the `SYSDBA` privilege. Decide whether `BERNST` should still be granted the `SYSDBA` privilege.

4. **Mandatory step:** When you are finished reviewing the reports execute the **`lab_10_04_04.sql`** script to remove the incomplete and misconfigured items.

```
SQL> set echo off
SQL> @/home/oracle/labs/lab_10_04_04.sql

Connected.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
Commit complete.
Connected.
Table dropped.
Revoke succeeded.
Revoke succeeded.
```

```
Revoke succeeded.
Grant succeeded.
Connected.
Revoke succeeded.
Revoke succeeded.
$
```

Practice 10-5: Viewing Videos

Overview

In this optional practice, you can review videos about auditing Database Vault operations unless your instructor just demonstrated the equivalent steps.

Note: Currently most OU classrooms do not have audio. To see and hear these videos with sound use the Oracle Learning Library: **www.oracle.com/goto/oll** and search for `Database Vault 12c`.

Tasks

1. To see how to audit Database Vault operations in a mixed auditing mode, you can view the following video:
   a. Double-click the **demos** folder and then the **dv03_m_audit** folder.
   b. In an Oracle classroom, double-click the HTML version (`dv03_m_audit.html`) to start the video. In other environments, you may use the MP4 version of the same video.
2. To see how to audit Database Vault operations in a unified auditing mode, you can view the following video:
   a. Double-click the **demos** folder and then the **dv04_u_audit** folder.
   b. In an Oracle classroom, double-click the HTML version (`dv04_u_audit.html`) to start the video. In other environments, you may use the MP4 version of the same video.

Practices for Lesson 11: Implementing Best Practices

Chapter 11

Practices for Lesson 11: Overview

Practices Overview

This practice is structured in the form of a workshop. You are presented with a scenario that involves implementing Database Vault restrictions to solve a problem or meet a requirement. The suggested approach for this practice is to read the scenario that is presented, and attempt to complete the tasks involved by using what you have learned so far in this course.

However, this practice does not present a complete solution. It is designed only to give you an opportunity to implement what you have learned and stimulate ideas for implementing Database Vault in your own environment.

You are encouraged to think beyond the scenario and tasks presented and to consider other options and ideas. If you have time, consider the business scenario here and create applicable separation-of-duty and application-protection matrices.

If you require assistance with this practice, the steps and details to complete this practice are provided on the pages that follow the scenario information.

The Scenario

Your company has hired an internal auditor to periodically check the IT systems to make independent audits happen more smoothly. In this practice, you look at some common situations that may need correcting to meet the audit requirements. Assume that your company has an HR application that is supported by tables owned by the **HR** schema, and an Order Entry application that is supported by tables owned by the **OE** schema. Some of the database users in your environment include:

- **SMAVRIS**: HR Manager
- **WSMITH**: HR Clerk
- **KPARTNER**: HR Clerk
- **BERNST**: HR Application DBA (has the **HR_DBA** role)
- **AHUNOLD**: Order Entry Application DBA (has the **OE_DBA** role)

You have implemented some restrictions by using Database Vault, but there are still some areas for improvement. Keep this in mind as you go through this practice.

The auditor has raised some questions and/or concerns. You need to answer the questions and, if necessary, implement solutions to the problems raised by the audit. Here are some questions from the auditor:

1. Are there database users who have the **SELECT ANY TABLE** privilege? Protect the data from such users by ensuring that the **HR** and **OE** schemas are protected. You have two application DBAs (**BERNST** for the **HR** data and **AHUNOLD** for the **OE** data). Protect each application schema by ensuring that each application DBA has access only to the tables for his or her application.
2. Currently there are restrictions in place for the **HR** DBA, which restricts his or her DBA activities to only after business hours unless he or she is connecting from the local machine. (The Domain factor of the **HR** DBA's local machine is considered `SECURE`.) This restriction should also be implemented for the **OE** DBA. (The Domain factor of the `OE` DBA's local machine is also considered `SECURE`.)
3. Are there controls in place to restrict who can become an application DBA? The auditor suggests restricting the usage of these roles to only where the Action for the session is **ADMIN**. It is recommended that you revoke the application roles from all users, and, instead, define new secure application roles that have a rule set that can implement the requirements of the auditor. Then lock down the **HR_DBA** and **OE_DBA** roles so that they can no longer be granted to anyone without the intervention of the **leo_dvowner** user.
4. The auditor is concerned about how well you are guarded against the accidental loss of data. Along these lines, there are certain commands that you want to control the use of, and one of them is **DROP TABLE**. The auditor recommends that you implement safeguards against this command executing accidentally. A suggestion is to disable the usage of the **DROP TABLE** command. The Database Vault owner could then enable the command on an as-needed basis.

   Note: Although one could use triggers to provide the same functionality, using Database Vault is the recommended approach because it conforms to the "separation of duties" concept (making the Database Vault Administrator responsible for this control, rather than the DBA).

5. Optional Challenge

Your DBA has come to you and needs to run the following command:

```
alter system dump datafile 5 block min 50 block max 55;
```

He or she receives an insufficient privileges error. This is just a one-time activity and he or she needs only temporary access to run this command. Suggest one possible way in which you can fix this problem so that he can run the command?

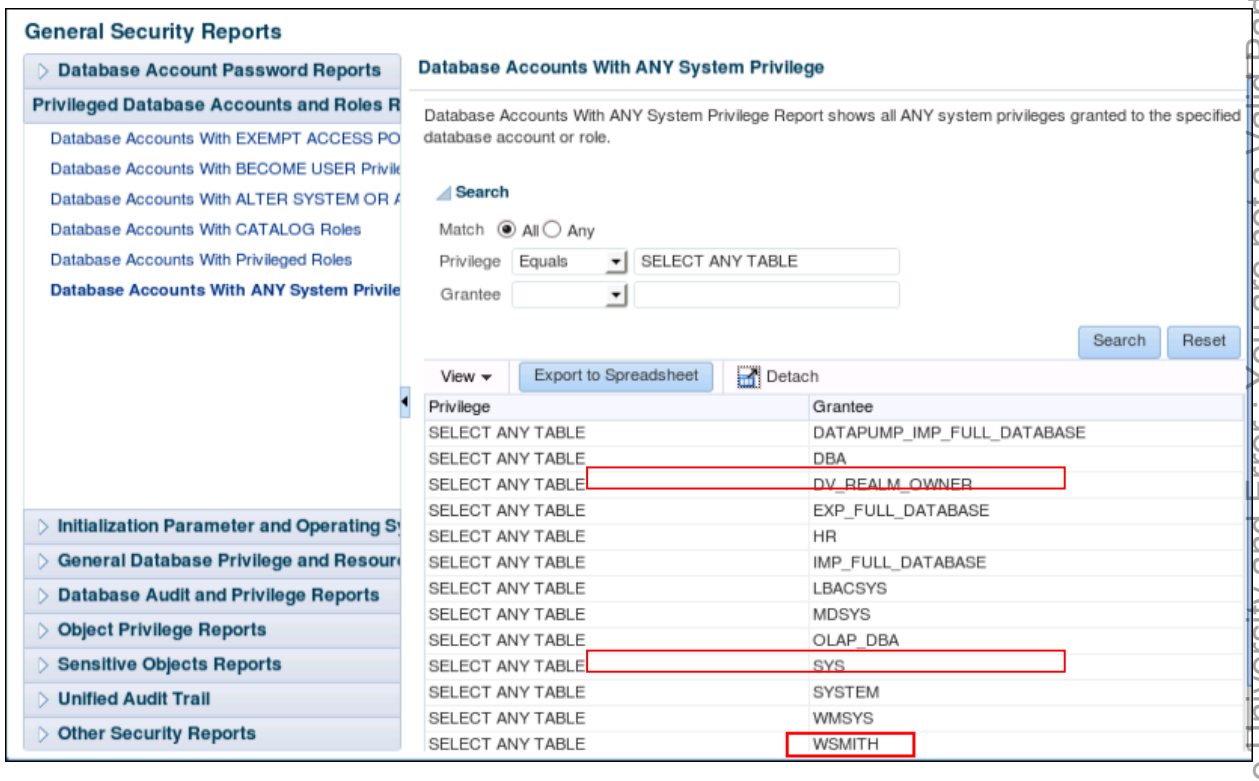
Hint: The answer is not as complicated as you might think.

Practice 11-1: Protecting Data from `SELECT ANY TABLE` Access

The auditor has asked you to show him or her the database users that have the **`SELECT ANY TABLE`** privilege. Run a report to show that information. Based on the results, ensure that both the **`HR`** and **`OE`** schemas are protected by realms.

Note: You should already have an **`HR_Schema`** realm from a previous practice. If not, refer back to Practice 5 and create it.

1. Log on to EM Cloud Control as the `leo_dvowner` user and run the System Privileges By Privilege report for the **`SELECT ANY TABLE`** privilege. This report can be found by navigating to Security > Reports.
   a. Click the **General Security Reports** tab. Click the **Privileged Database Accounts and Roles Reports**, and then click the **Database Accounts With ANY System Privilege**. In the Privilege fields, select or enter `Equals` and `SELECT ANY TABLE` and click **Search**.



View the results. Based on the report, you (and the auditor) can see that there are users who can select any table in the database even if they do not need to.

2. To protect the data, each individual schema should be protected by a realm. You already have a realm for the **HR** schema and the **HR** DBA has access to this realm. You now need to create a realm for the **OE** schema and allow the **OE** DBA to have access to the **OE** objects in the realm.
   a. Navigate to Database Vault > Administration > Realms page. Click **Create**.
   a. On the Create Realm page, enter the following information and click **Next**:
      Name: **OE_Schema**
      Description: **A realm to protect the OE schema objects.**
   b. In the Realm Secured Objects region, click **Add**.
   c. On the Create Realm Secured Object page, enter the following information and click **OK** and then **Next**.
      Object Owner: **OE**
      Object Type: **%**
      Object Name: **%**
   d. In the Realm Authorizations region, click **Add**. Select **AHUNOLD [USER]** as the Realm Authorization Grantee and click **OK**.
   e. Click **Done** and then **Finish** to finish creating the **OE_Schema** realm.
3. Prove to the auditor that the users with the **SELECT ANY TABLE** privilege cannot read realm-protected tables outside of their realm.
   a. As **AHUNOLD**, attempt to SELECT from the **HR.EMPLOYEES** table.

```
SQL> CONNECT AHUNOLD
Enter password:
Connected.
SQL> SELECT last_name, salary FROM HR.EMPLOYEES
     WHERE EMPLOYEE_ID = 108;
SELECT last_name, salary FROM HR.EMPLOYEES
                              *
ERROR at line 1:
ORA-01031: insufficient privileges


SQL>
```

b. As **BERNST**, attempt to SELECT from the **OE.ORDERS** table.

```
SQL> connect BERNST
Enter password:
Connected.
SQL> SELECT customer_id, order_total FROM OE.ORDERS
     WHERE ROWNUM < 10;
SELECT customer_id, order_total FROM OE.ORDERS
                                       *
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
```

Practice 11-2: Restricting OE DBA Activities to Nonbusiness Hours

The auditor has seen that you have the **HR** DBA restricted to performing DBA activities after the regular business hours unless he or she connects from the local machine. (The Domain factor of the **HR** DBA's local machine is considered SECURE.) This restriction should also be implemented for the **OE** DBA. (The Domain factor of the OE DBA's local machine is also considered SECURE.)

1. Use the Non_Work_Hours rule set to restrict DBA activity on the **OE** schema to only the hours defined by the rule set. To do this, edit the **HR_Schema** realm and add the Non_Work_Hours rule set as an authorization rule set for the **AHUNOLD** participant.
   a. As the leo_dvowner user, use Enterprise Manager Cloud Control and navigate to Oracle Database Vault > Administration > Realms.
   b. Select the **HR_Schema** realm and click **Edit**.
   c. In the Realm Authorizations page, add **AHUNOLD** as a participant with Realm Authorization Rule Set to Non_Work_Hours and click **OK** twice.
   d. On the Edit Realm: HR_Schema: Realm Authorizations page, click **Done**.
   e. On the Edit Realm: HR_Schema: Review page, click **Finish**.
2. Confirm that **AHUNOLD** can do his DBA tasks only during nonwork hours by attempting to create a new table in the **OE** schema. Confirm your system time.

   Try connecting with your **orcl** service name and creating a table in the **OE** schema. This works because the IP address for the connection is mapped to the **SECURE** identity for the Domain factor. (Refer to Practice 5-1 for more details.)

```
SQL> !date
Thu Jul 10 11:23:40 UTC 2014

SQL>
SQL> connect ahunold
Enter password:
Connected.
SQL> create table oe.x (a number);
create table oe.x (a number)
*
ERROR at line 1:
ORA-47401: Realm violation for CREATE TABLE on OE.X

SQL> connect ahunold@orcl
Enter password:
Connected.
SQL> create table oe.x (a number);

Table created.

SQL> drop table oe.x;
```

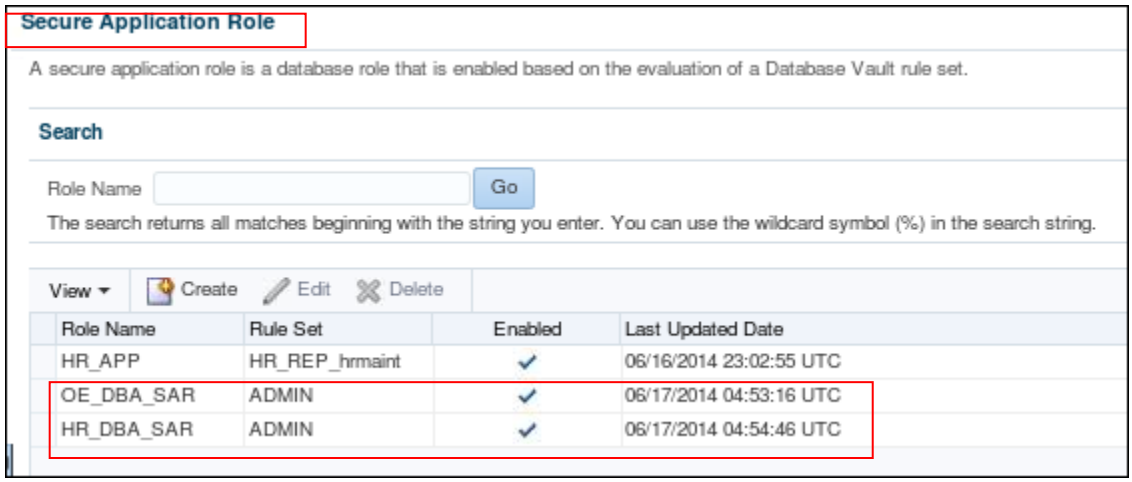
```
Table dropped.

SQL>
```

Practice 11-3: Locking Down the DBA Roles

The auditor realizes that the application DBAs are now locked into their realms, but wants further control on who can become an application DBA. You need to restrict the usage of these roles to only where Action for the session is **ADMIN**. You decide to revoke the application roles from all users, and, instead, define new secure application roles that have a rule set that can implement the requirements of the auditor. Then lock down the **HR_DBA** and **OE_DBA** roles so that they can no longer be granted to anyone without the intervention of the **leo_dvowner** user.

1. Create a factor called **ACTION** that represents the **ACTION** context value for the session.
   a. Navigate to Database Vault > Administration > Factors page, and click **Create**.
   b. Enter the following on the Create Factor: General page, and click **Next**:
      Name: **ACTION**
      Description: **Represents the ACTION context value for a session.**
      Factor Type: **Application**
   c. Enter the following on the Create Factor: Configurations page, and click **Next**:
      Factor Identification: **By Method**
      Evaluation: **By Access**
      Factor Labeling: **By Self**
      Retrieval Method: **sys_context('USERENV','ACTION')**
   d. Click **Done** and then **Finish** to complete the factor creation.
2. Create a rule set called **ADMIN** that ensures that the **ACTION** factor is equal to **ADMIN**.
   a. Navigate to Database Vault > Administration > Rule Sets page, and click **Create**.
   b. Enter the following on the ”Create Rule Set: General” page and click **Next**:
      Name: **ADMIN**
      Description: **Rule set to ensure ACTION factor is ADMIN.**
   c. Click **Create Rule** on the “Create Rule Set: Associate with Rules” page.
   d. Enter the following on the “Create Rule Set: Associate with Rules” and click **OK**:
      Name: **ADMIN_ACTION**
      Rule Expression: **dvf.f$action = 'ADMIN'**
   e. Click **Done** and then **Finish** to complete the **ADMIN** rule set creation.
3. Create the **OE_DBA_SAR** and **HR_DBA_SAR** secure application roles. The rule set to be satisfied for each of these is **ADMIN**.
   a. Navigate to Database Vault > Administration > Secure Application Roles page, and click **Create**.
   b. Name the role **OE_DBA_SAR**, and set Rule Set to **ADMIN**. Click **OK**.
   c. Click **OK** to complete the **OE_DBA_SAR** secure application roles creation.
   d. Click **Create** again to create the other role, **HR_DBA_SAR**.
   e. Set Rule Set to **ADMIN**, click OK and click **OK**, to complete the **HR_DBA_SAR** secure application roles creation.

f. Review the two new secure application roles.



4. As DBA_PSMITH, revoke the **OE_DBA** and **HR_DBA** roles from all users that have them.

```
SQL> CONNECT DBA_PSMITH
Enter password:
Connected.
SQL> revoke OE_DBA from AHUNOLD;

Revoke succeeded.

SQL> revoke HR_DBA from BERNST;

Revoke succeeded.

SQL>
```

5. Grant the **OE_DBA** and **HR_DBA** roles to the new secure application roles, respectively.

```
SQL> grant OE_DBA to OE_DBA_SAR;

Grant succeeded.

SQL> grant HR_DBA to HR_DBA_SAR;

Grant succeeded.

SQL>
```

6. Test the new secure application roles by logging in to SQL*Plus. Remember to connect by using your service name so that you satisfy one of the rules in the Non_Work_Hours rule set and, therefore, are able to perform your DBA task. Attempt to create a table in the appropriate schema (OE for AHUNOLD and HR for BERNST). What happens?

   Then set the Action application information variable to **ADMIN** and enable the appropriate role (OE_DBA_SAR for AHUNOLD and HR_DBA_SAR for BERNST). Test again to see whether

you can create the same table. What happens? Do this for the **BERNST** user and for the **AHUNOLD** user.

a. Enter the following in the SQL*Plus session:

```
SQL> connect ahunold@orcl
Enter password:
Connected.
SQL> create table oe.mytab (a number);
create table oe.mytab (a number)
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL>
```

b. Enter the following to set **ACTION** to **ADMIN**:

```
SQL> EXECUTE DBMS_APPLICATION_INFO.SET_ACTION('ADMIN')

PL/SQL procedure successfully completed.

SQL> select sys_context('USERENV','ACTION') from dual;

SYS_CONTEXT('USERENV','ACTION')
----------------------------------------------------------
ADMIN

SQL>
```

c. Enter the following to set the secure application role:

```
SQL> EXECUTE DVSYS.DBMS_MACSEC_ROLES.SET_ROLE('OE_DBA_SAR')

PL/SQL procedure successfully completed.

SQL>
```

d. Attempt to create the **OE.MYTAB** table again. Drop the table when you finish.

```
SQL> create table oe.mytab (a number);

Table created.

SQL> drop table oe.mytab;

Table dropped.

SQL>
```

e. Repeat for the **BERNST** user, attempting to create an **HR.MYTAB** table. Enter the following:

```
SQL> connect bernst@orcl
Enter password:
Connected.
SQL> create table hr.mytab (a number);
create table hr.mytab (a number)
*
ERROR at line 1:
ORA-01031: insufficient privileges

SQL> EXECUTE DBMS_APPLICATION_INFO.SET_ACTION('ADMIN')

PL/SQL procedure successfully completed.

SQL> EXECUTE DVSYS.DBMS_MACSEC_ROLES.SET_ROLE('HR_DBA_SAR')

PL/SQL procedure successfully completed.
```

```
SQL> select sys_context('USERENV','ACTION') from dual;

SYS_CONTEXT('USERENV','ACTION')
----------------------------------------------------------------
ADMIN
```

```
SQL> create table hr.mytab (a number);

Table created.

SQL> drop table hr.mytab;

Table dropped.

SQL>
```

7. Lock down the **HR_DBA** and **OE_DBA** roles so that they can no longer be granted to anyone without the intervention of **leo_dvowner**.
   a. Navigate to Database Vault > Administration > Realms page, and click **Create**.
   b. On the Create Realm: General page, enter the following and click **Next**:
      Name: **APP_DBA**
      Description: **A realm to protect the application DBA roles.**

c. Click **Add** on the Create Realm: Realm Secured Objects page, and enter the following:

Object Owner: **ANONYMOUS**

Object Type: **ROLE**

Object Name: **HR_DBA**

Note: Roles do not have owners. Therefore, in this case, the value for the Object Owner field is ANONYMOUS.

d. Still on the Create Realm: Realm Secured Objects page, click **Add** to enter the following:

Object Owner: **ANONYMOUS**

Object Type: **ROLE**

Object Name: **OE_DBA**

e. Review the two secured roles in this realm and click **Done** and then **Finish** to complete the creation of this realm.

Now that these roles are in a realm that has no Owners set, no user can grant these roles unless leo_dvowner disables this realm or removes the roles from it.

8. Test to illustrate that these two roles can no longer be granted. Connect as **SYSTEM** or any other user that has the **GRANT ANY ROLE** privilege and attempt to grant the protected roles to **SMAVRIS** by entering the following in the SQL*Plus session:

```
SQL> CONNECT DBA_PSMITH
Enter password:
Connected.
SQL> GRANT HR_DBA to SMAVRIS;
GRANT HR_DBA to SMAVRIS
*
ERROR at line 1:
ORA-47410: Realm violation for GRANT on HR_DBA

SQL> GRANT OE_DBA to SMAVRIS;
GRANT OE_DBA to SMAVRIS
*
ERROR at line 1:
ORA-47410: Realm violation for GRANT on OE_DBA

SQL>
```

Practice 11-4: Preventing Data Loss

You understand that in addition to preventing intentional, unwanted data manipulation, you also need to guard against accidental loss of data. There are certain commands that you want to control the use of, and one of them is **DROP TABLE**. You need to implement safeguards against this command executing accidentally, which is a practice one may follow in a production database. In this practice, you set up a command rule in order to perform a **DROP TABLE**.

1. Create the **DROP TABLE** command rule that has `Disabled` as its rule set. Distinguish the `Disabled` rule set from the `Disabled` status.
   a. Navigate to Database Vault > Administration > Command Rules page, and click **Create**.
   b. Choose the following from the drop-down lists on the Create Command Rule page and click **OK**:

      Command: **DROP TABLE**

      Rule Set: **Disabled**

2. Test the command rule by creating and then dropping a table. Use the **SH** user for these tests. First, unlock the **SH** user, then create a test table, and then attempt to drop it.

   Note: The password for the **SH** user is **oracle_4U.**

```
SQL> conn bea_dvacctmgr
Enter password:
Connected.
SQL> alter user sh identified by oracle_4U account unlock;

User altered.

SQL> connect sh
Enter password:
Connected.
SQL> create table y (a int);

Table created.


SQL> drop table y;
drop table y
*
ERROR at line 1:
```

```
ORA-47400: Command Rule violation for DROP TABLE on SH.Y

SQL>
```

3. If you ever require the **DROP TABLE** command to work again, the **leo_dvowner** user can simply set the rule set for the **DROP TABLE** command rule to Enabled. Test this functionality now.
   a. Navigate to Database Vault > Administration > Command Rules page, select the **DROP TABLE** command rule that you just created, and click **Edit**.
   b. Set Rule Set to Enabled, and then click **OK**.
   c. Still as the SH user, issue the following at SQL prompt to reattempt the **DROP TABLE** command:

```
SQL> drop table y;

Table dropped.

SQL>
```

Practice 11-5: Allowing Temporary `ALTER SYSTEM` Command Access

As stated earlier, your DBA has come to you and needs to run the following command:

```
alter system dump datafile 5 block min 50 block max 55;
```

He or she receives an insufficient privileges error. This is just a one-time activity and he or she needs only temporary access to run this command. Suggest one possible way in which you can fix this problem so that the DBA can run the command?

1. Run the command and see the error that occurs. Log in as `DBA_PSMITH` or as `sys` and attempt to execute the command:

```
SQL> connect / as sysdba
Connected.
SQL> alter system dump datafile 5 block min 50 block max 55;
alter system dump datafile 5 block min 50 block max 55
*
ERROR at line 1:
ORA-01031: insufficient privileges


SQL>
```

2. One way is to disable the command rule for the `ALTER SYSTEM` command. This command rule comes preconfigured with Oracle Database Vault 12*c*.
   a. Navigate to Database Vault > Administration > Command Rules page.
   b. Click **Show Oracle defined Command Rules** to view the list of preconfigured command rules.
   c. Select the **`ALTER SYSTEM`** command rule and click **Edit**.
   d. Select **Disabled** for Status and click **OK**.
3. Return to the SQL*Plus session and run the command again (still connected as `DBA_PSMITH`).

```
SQL> alter system dump datafile 5 block min 50 block max 55;


System altered.


SQL> EXIT
$
```

4. Re-enable the `ALTER SYSTEM` command rule again.
   a. Navigate to Database Vault > Administration > Command Rules page.
   b. Click **Show Oracle defined Command Rules** to view the list of preconfigured command rules.
   c. Select the **`Alter System`** command rule and click **Edit**.
   d. Select **Enabled** for Status and click **OK**.