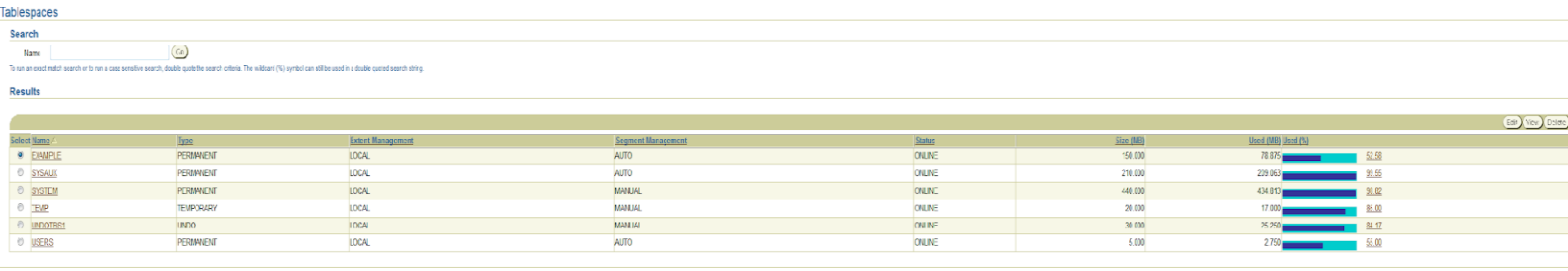
# *Administration d'un serveur Oracle*

## Contenu

## 3. *Managing tablespaces*

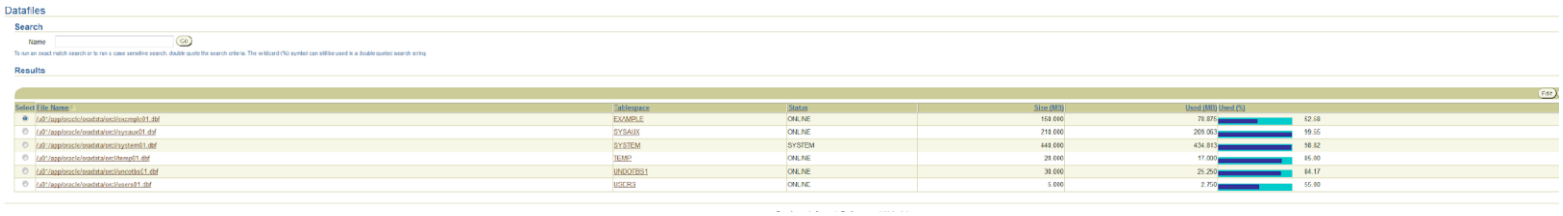
### 1)

-Use Database Control to view all tablespaces in your database. For each tablespace, record the tablespace name, type, size and percent used.




### 2)

-View all datafiles in your database. For each datafile record the file name, tablespace name, current size, autoextend status, and maximum file size (if autoextend is enabled).




-The SYSTEM tablespace is over 90 % full. Based on the information you've just collected, should you be concerned?

Le tablespace SYSTEM contient le datafile /u01/app/oracle/oradata/orcl/system01.dbf, qui est lui même rempli à 98,82 %.
Au vu des infos données dans la partie "background", qui explique que les tablespaces contiennent les données pour les applications, on peut supposer que le tablespace SYSTEM contient les données système de Oracle, et qui il a été fait afin de juste contenir les informations nécessaires, d'où le fait que il soit quasiment plein.

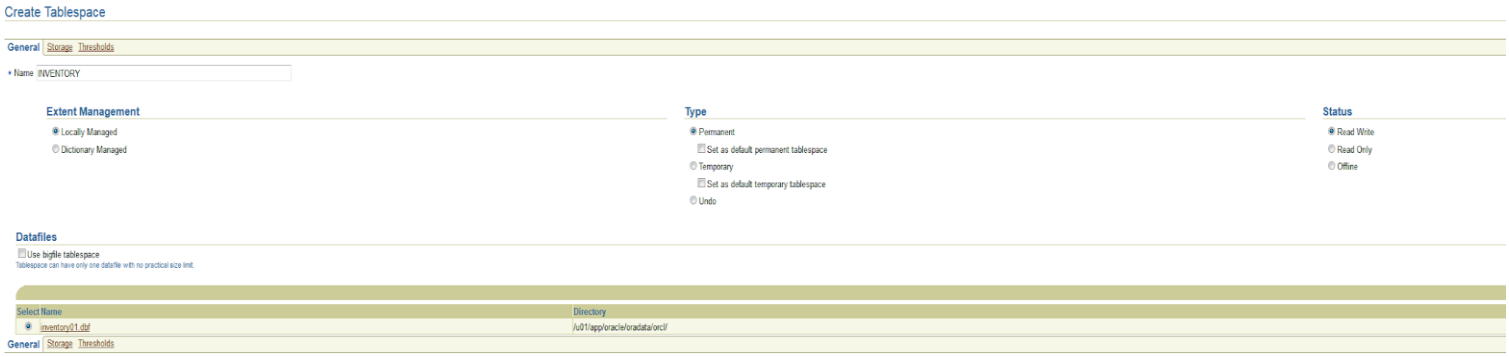
De ce fait, il n'est pas nécessaire de s'inquiéter concernant le fait qu'il soit rempli à 90%.

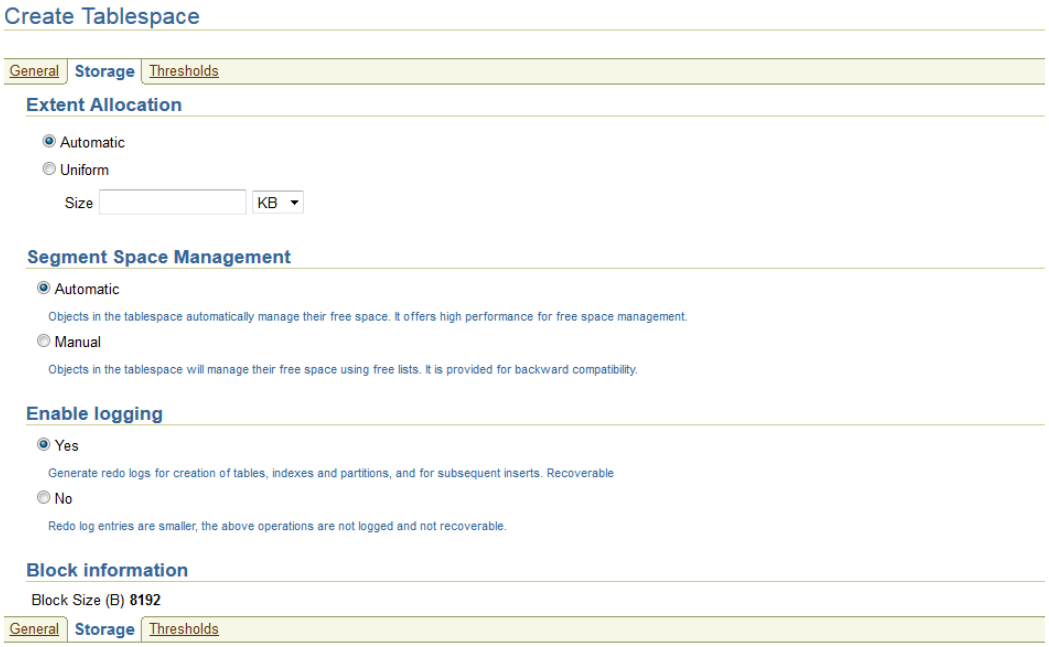
-Why is autoextend an attribute of the data file rather than the tablespace?

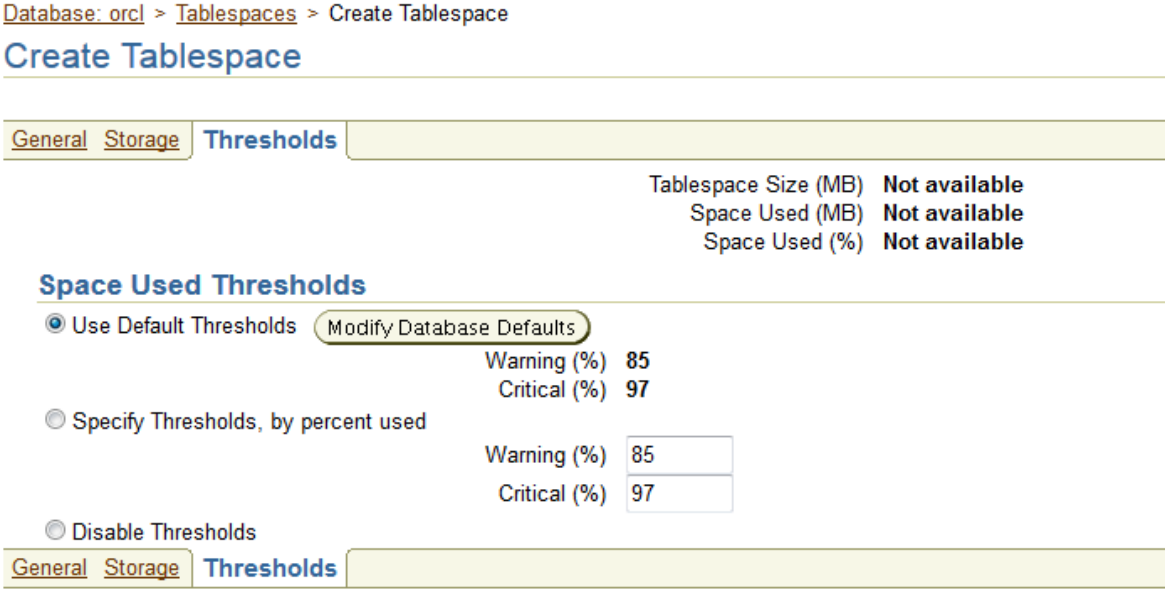
Le datafile contient les données, c'est donc lui qui va augmenter au fur et à mesure que il a des nouveaux ajouts. Quand le fichier à besoin de place, le datafile augmente et le tablespace suis.

**3)**

-Create a new tablespace to hold information for the inventory application.



| Select | Name | Type | Extent Management | Segment Management | Status | Size (MB) | Used (MB) | Used (%) |
|---|---|---|---|---|---|---|---|---|
| ◉ | EXAMPLE | PERMANENT | LOCAL | AUTO | ONLINE | 150.000 | 78.875 | 52.58 |
| ○ | INVENTORY | PERMANENT | LOCAL | AUTO | ONLINE | 50.000 | .063 | 0.13 |
| ○ | SYSAUX | PERMANENT | LOCAL | AUTO | ONLINE | 210.000 | 209.063 | 99.55 |
| ○ | SYSTEM | PERMANENT | LOCAL | MANUAL | ONLINE | 440.000 | 434.875 | 98.84 |
| ○ | TEMP | TEMPORARY | LOCAL | MANUAL | ONLINE | 20.000 | 17.000 | 85.00 |
| ○ | UNDOTBS1 | UNDO | LOCAL | MANUAL | ONLINE | 30.000 | 25.313 | 84.38 |
| ○ | USERS | PERMANENT | LOCAL | AUTO | ONLINE | 5.000 | 2.750 | 55.00 |

## 4. *Administrating Database Users*

**1)**

-Create a profile named **HRPROFILE** that limits idle time to 15 minutes. Leave all other fields set to **DEFAULT**.

General | Password

* Name HRPROFILE

**Details**

CPU/Session (Sec./100) DEFAULT

CPU/Call (Sec./100) DEFAULT

Connect Time (Minutes) DEFAULT

Idle Time (Minutes) 15

**Database Services**

Concurrent Sessions (Per User) DEFAULT

Reads/Session (Blocks) DEFAULT

Reads/Call (Blocks) DEFAULT

Private SGA (KBytes) DEFAULT

Composite Limit (Service Units) DEFAULT

General | Password

**2)**

-Set the initialization parameter **RESOURCE_LIMIT** to **TRUE** so that your profile limits will be enforced.

| Name | Help | Revisions | Value | Type |
|---|---|---|---|---|
| resource_limit | | | TRUE | Boolean |

**3)**

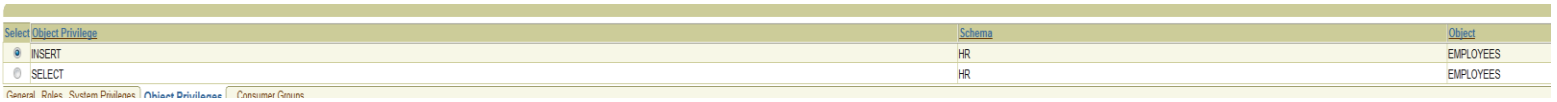
-Create a role named **HRCLERK** without authentication and with **SELECT** and **UPDATE** permissions on the hr.employees table. This role will be used for clerks of the HR department.

General | Roles | System Privileges | Object Privileges | Consumer Groups

* Name HRCLERCK

Authentication None

General | Roles | System Privileges | Object Privileges | Consumer Groups

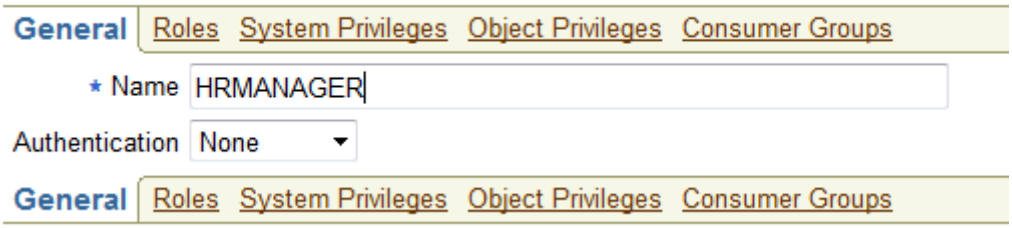


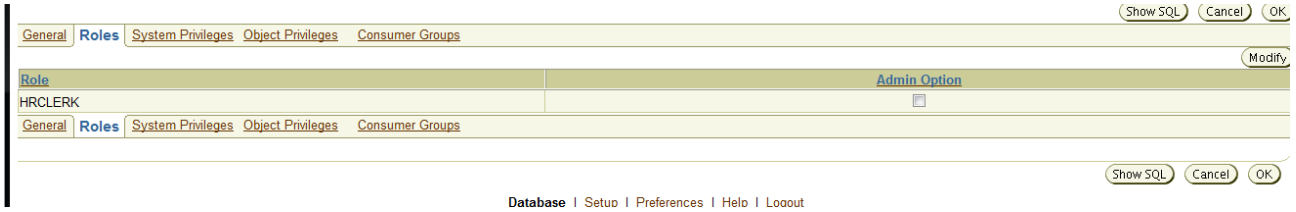
| Select | Object Privilege | Schema | Object |
|---|---|---|---|
| ◉ | INSERT | HR | EMPLOYEES |
| ○ | SELECT | HR | EMPLOYEES |

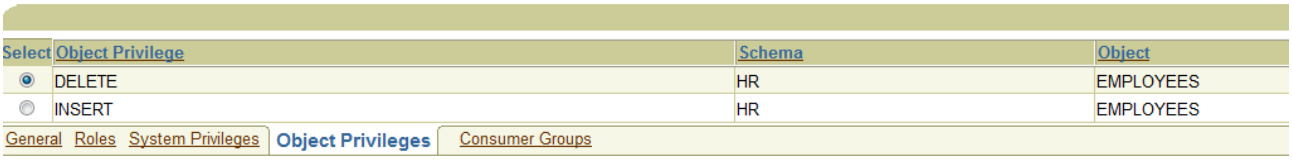
General Roles System Privileges Object Privileges Consumer Groups

**4)**

-Create a role named **HRMANAGER** with **INSERT** and **DELETE** permissions on the hr.employees table. Grant the **HRCLERK** role to the **HRMANAGER** role. This role will be used by managers of the HR department.



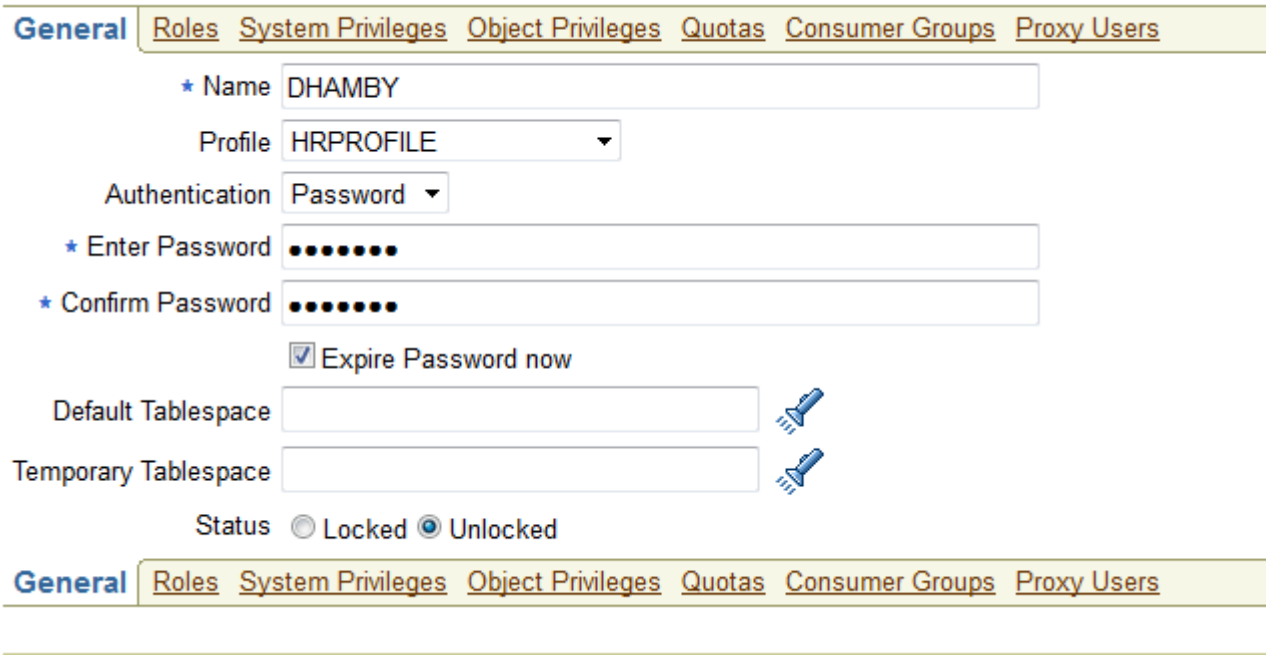



| Select | Object Privilege | Schema | Object |
|---|---|---|---|
| ◉ | DELETE | HR | EMPLOYEES |
| ○ | INSERT | HR | EMPLOYEES |

General Roles System Privileges Object Privileges Consumer Groups

**5)**

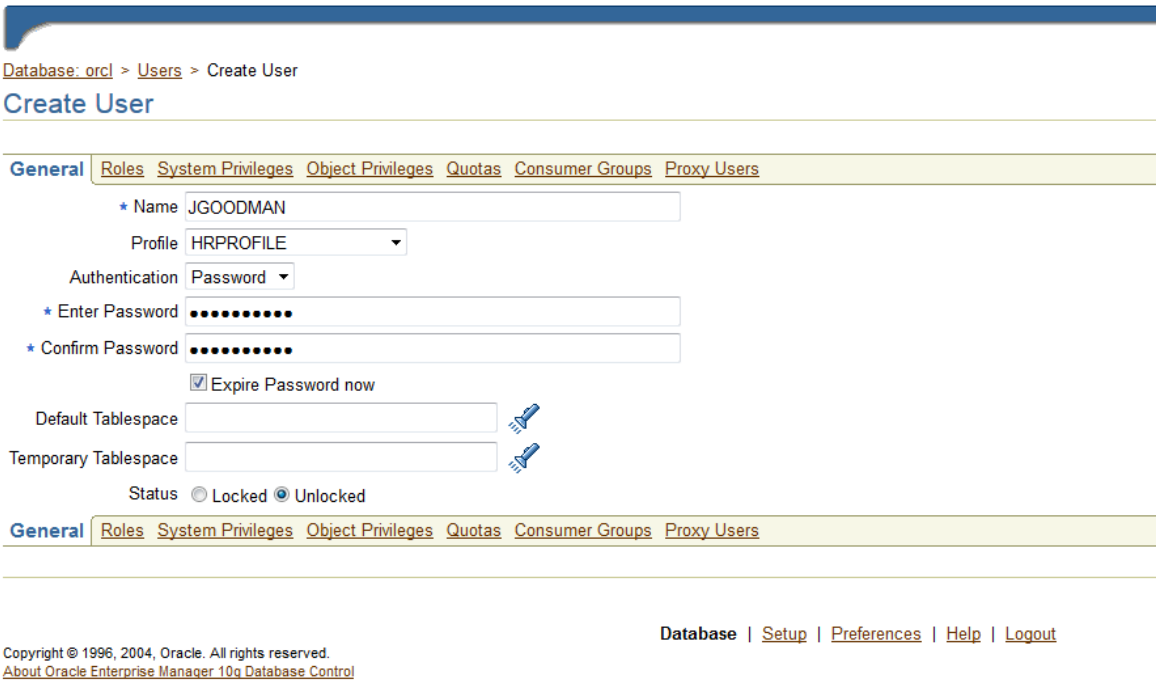
-Create an account for David Hamby (DHAMBY), a new HR clerk. His profile is **HRPROFILE**, his password is newuser and this one expires immediately.





**6)**

-Create an account for Jenny Goodman (JGOODMAN), the HR new manager. His profile is HRPROFILE, his password is newmanager and this one expire immediately.



**7)**

-Connect to the database as user DHAMBY using SQL*Plus. Attemps to select from the hr.employees table.

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


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

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> SELECT salary FROM hr.employees;

    SALARY
----------
     24000
     17000
     17000
      9000
      6000
      4800
      4800
      4200
```

**8)**

-New attempt to delete a record from the hr.employees table. You may get an error.

```
Elapsed: 00:00:00.00
SQL> DELETE FROM hr.employees
  2  WHERE salary = 8300;
DELETE FROM hr.employees
            *
ERROR at line 1:
ORA-01031: insufficient privileges
```

**9)**

-Connect to the database as JGOODMAN and attempt to select and then delete (employee_id = 143 for example) from the hr.employees table.

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


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

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Proc
With the Partitioning, OLAP and Data Mining options

SQL> DELETE FROM hr.employees
  2  WHERE employee_id = 143;

1 row deleted.
```

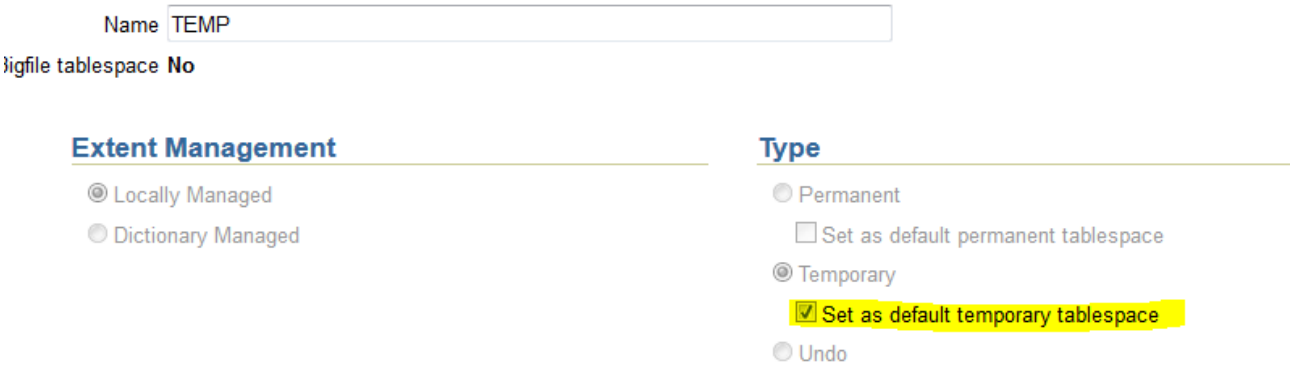
**10)**

-Roll back the delete operation because this was only a test.

```
SQL> ROLLBACK;

Rollback complete.

Elapsed: 00:00:00.00
SQL>
```

-When you created the new users you did not select a default temporary tablespace. What determines which tablespaces the new users will use?

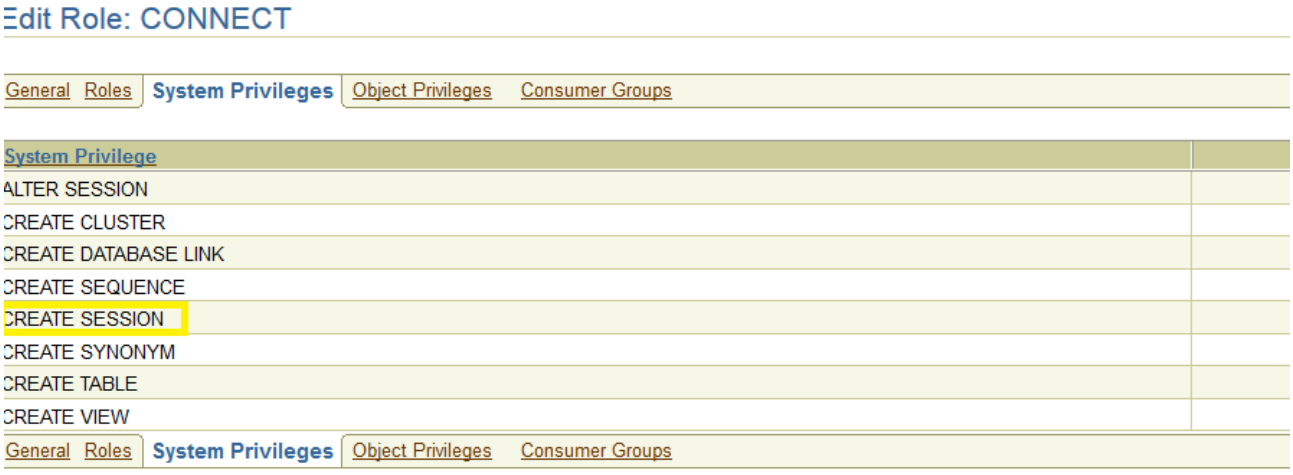
Dans le cas où nous avons pas décider d'un tablespace temporaire pour un utilisateur, Oracle va utiliser "TEMP", qui est la tablespace temporaire par défaut:




Comme on peut le voir, elle a été définie ainsi.

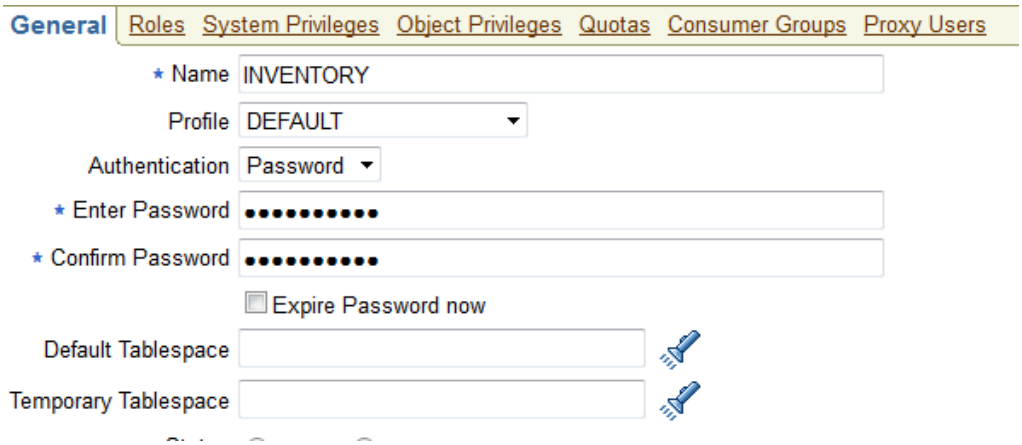
-You did not grant the **CREATE SESSION** system privilege to either of the new users, but they can both connect to the database. Why?

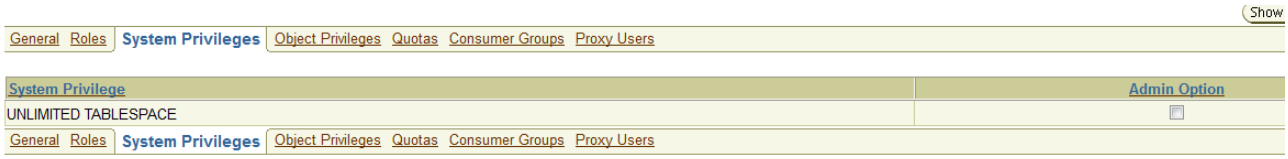
Quand nous créons un utilisateur, il obtient le rôle de "connect" par défaut.
Ce rôle possède un ensemble de privilèges, comme celui de "create session", par exemple.

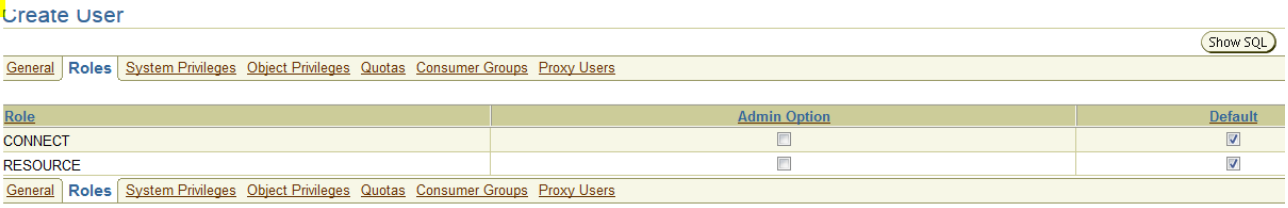



-Create a new user account to own database objects for a new inventory application. The username should be

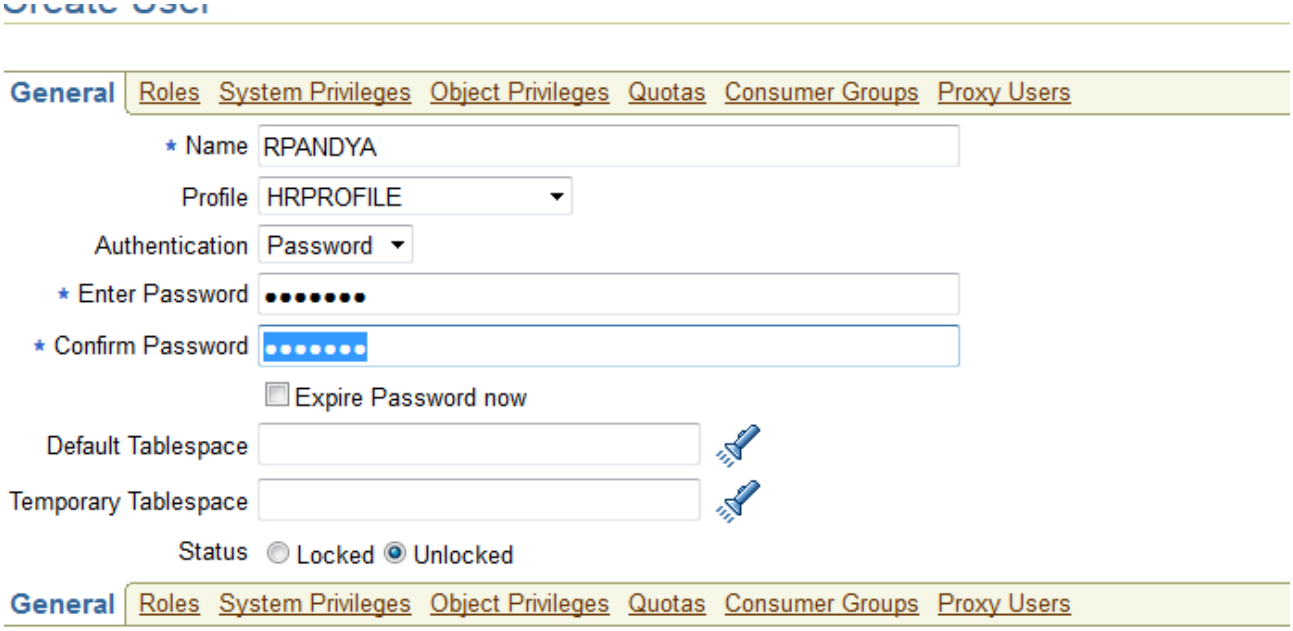
**INVENTORY** with a password of verysecure. Make the user's default tablespace the **INVENTORY** tablespace. Grant the user the **CONNECT** and **RESOURCE** role. Also give him the **UNLIMITED_TABLESPACE** system privilege.




-Leave one of the new users (RPANDYA) connected to the database during the next lesson. Verify that the user is automatically logged out after fifteen minutes.

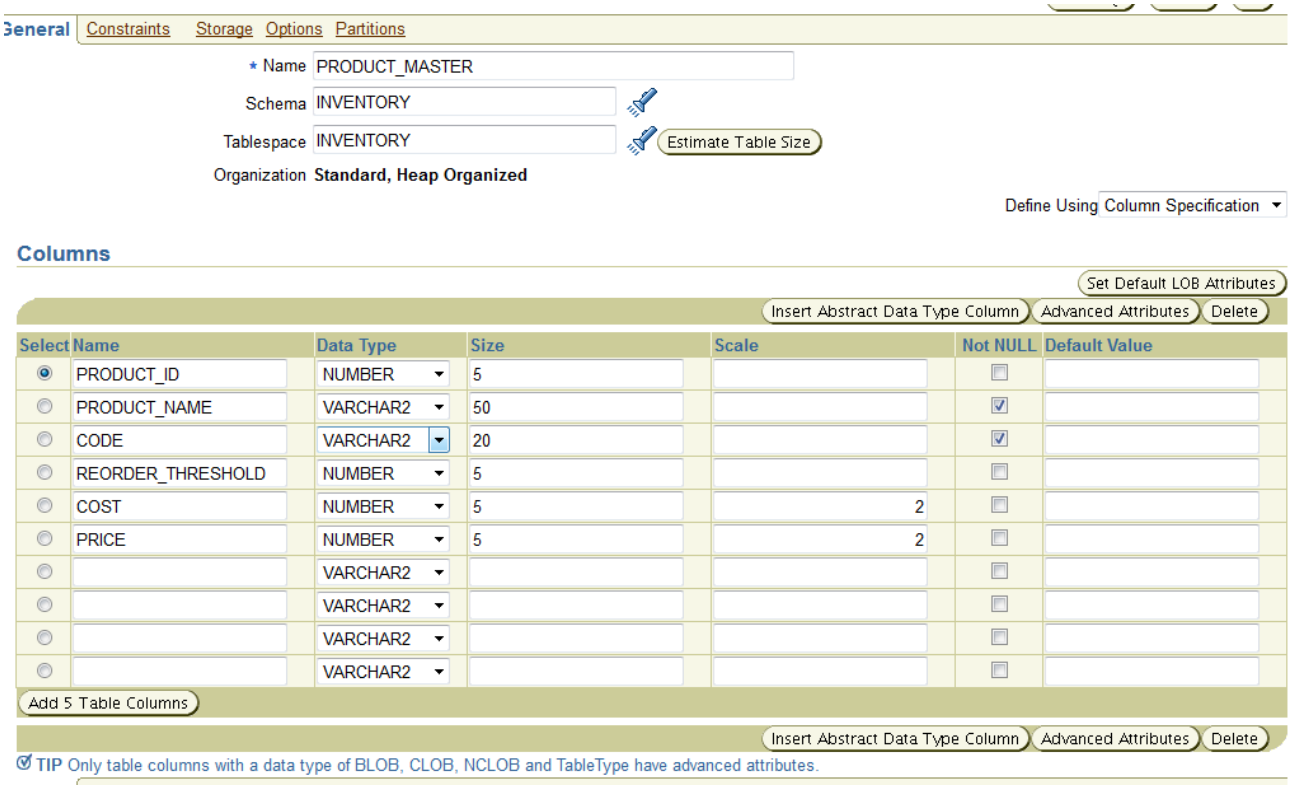


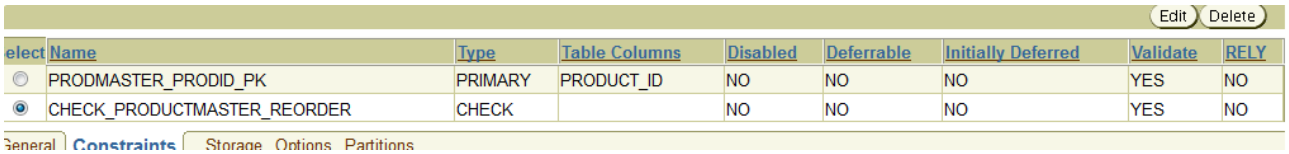
```
Enter user-name: RPANDYA
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL>
SQL>
SQL>
SQL>
SQL> exit^H^H^H^H
Usage: { EXIT | QUIT }  [ SUCCESS | FAILURE | WARNING | n |
       <variable> | :<bindvariable> ]  [ COMMIT | ROLLBACK ]
ERROR:
ORA-02396: exceeded maximum idle time, please connect again


Disconnected from Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options (with complications)
[oracle@vmware oracle]$
```

## 5. *Managing Schema Objects*

**1)**

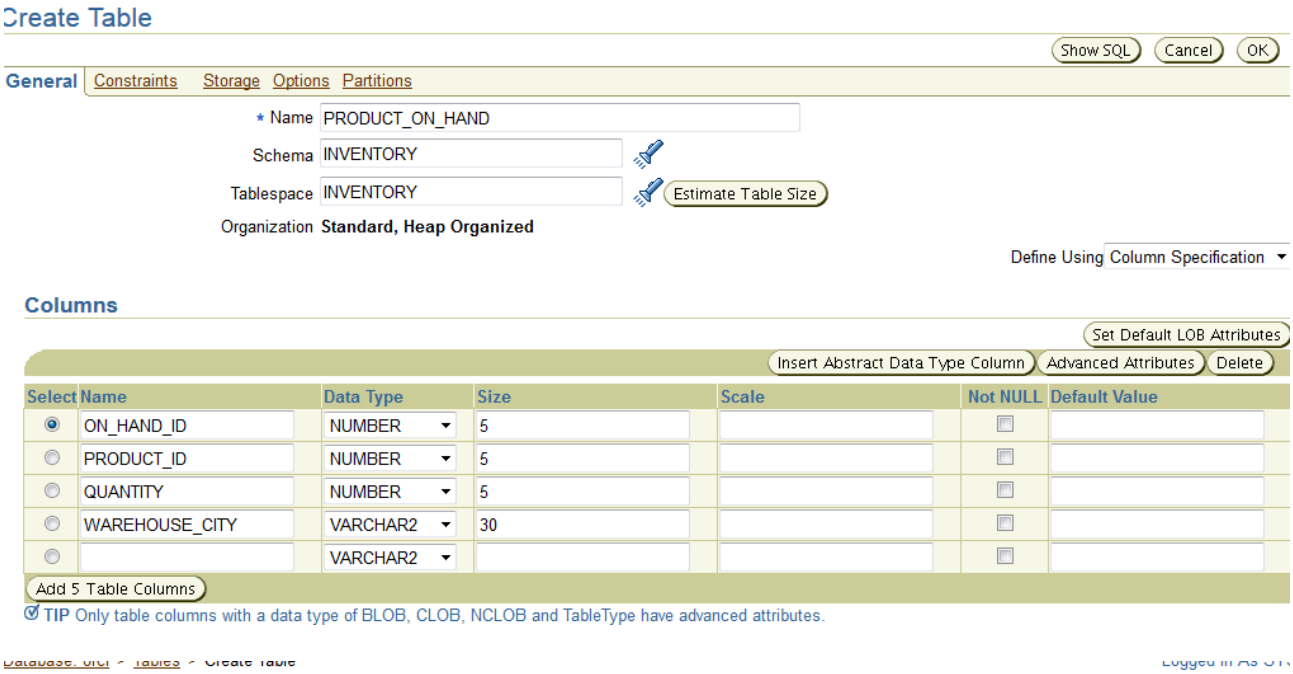
-In the **INVENTORY** tablespace, create the **PRODUCT_MASTER** table, in the **INVENTORY** schema, using Enterprise Manager.

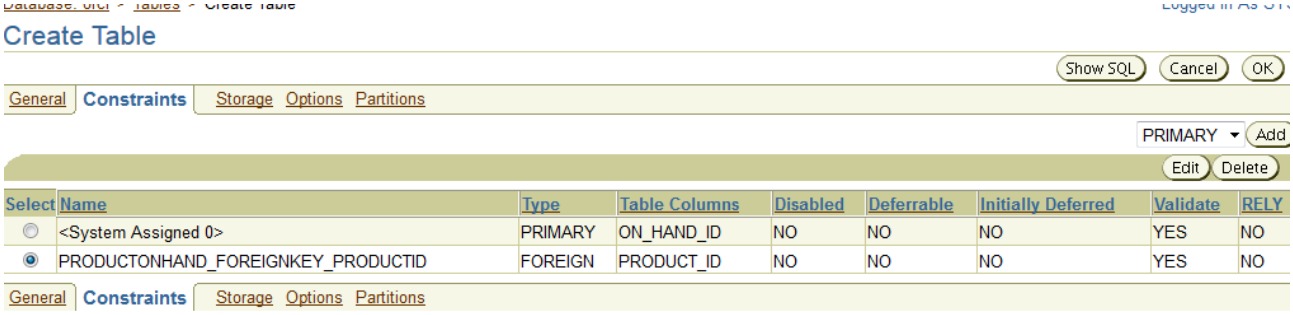


**2)**

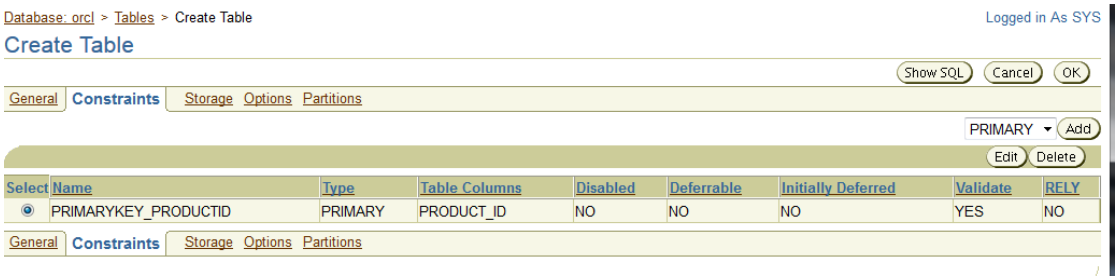
-In the **INVENTORY** tablespace, create the **PRODUCT_ON_HAND** table, in the **INVENTORY** schema.

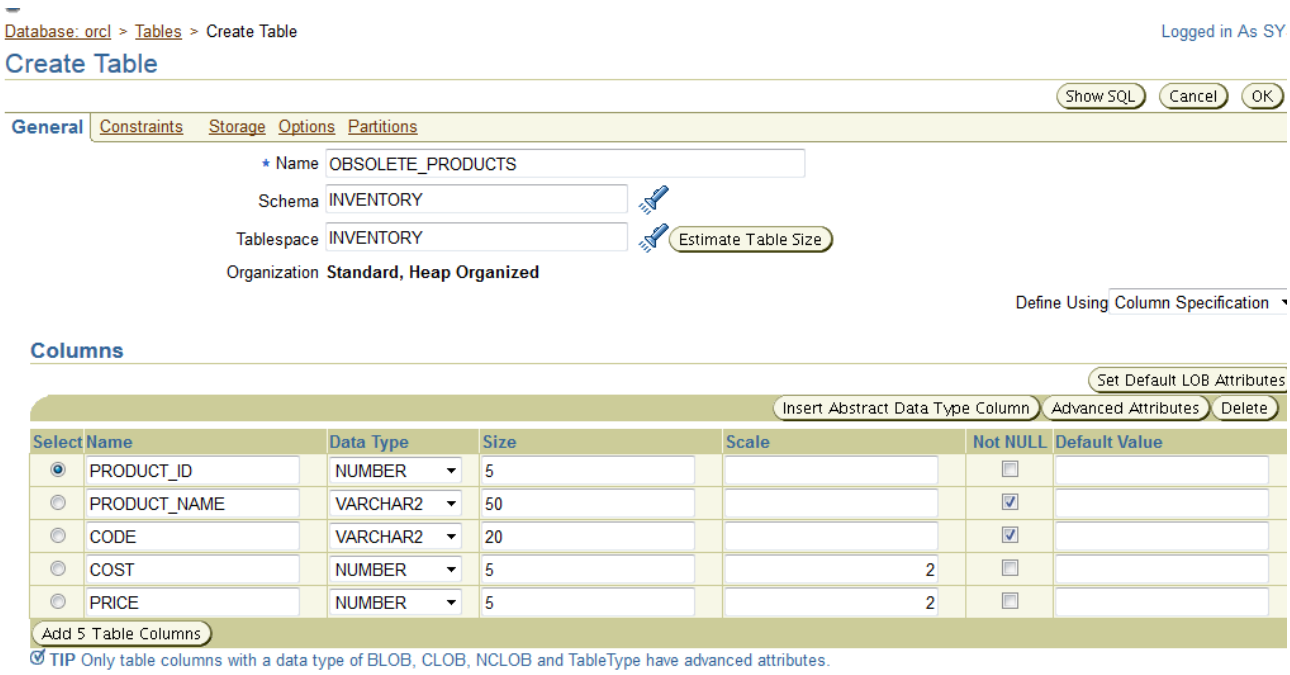



**3)**

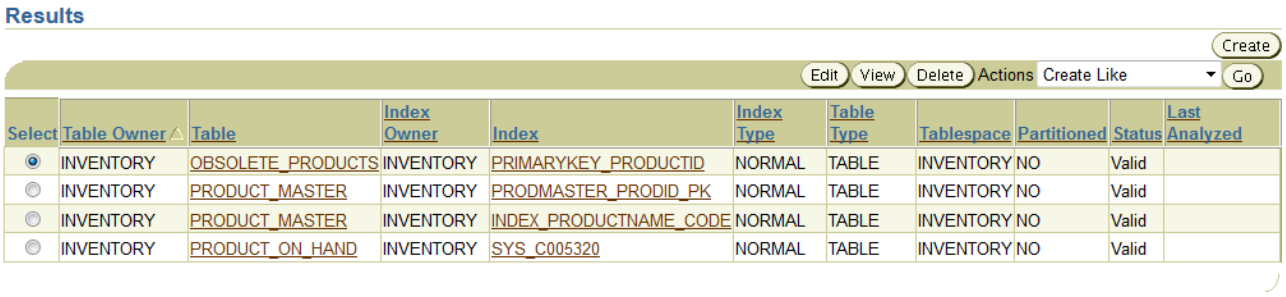
-Then create the OBSOLETE_PRODUCTS table.

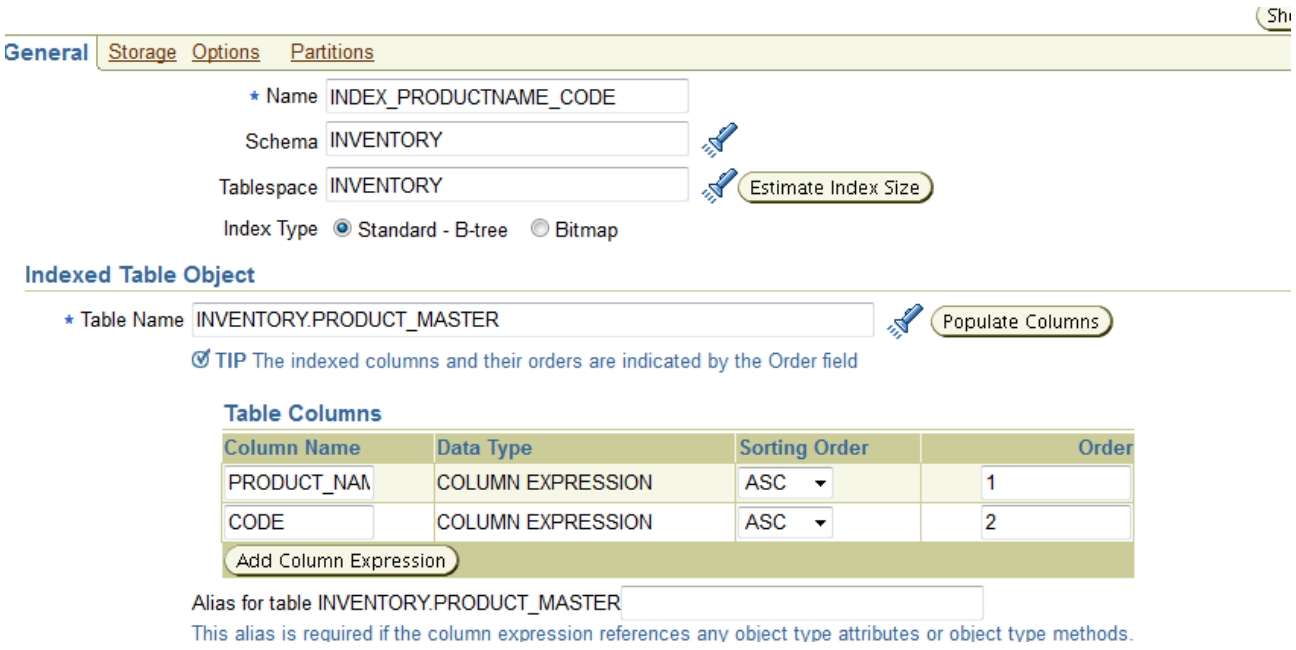





**4)**

-In the **INVENTORY** tablespace, create an index on the PRODUCT_NAME and CODE columns of the PRODUCT_MASTER table in the **INVENTORY** schema.



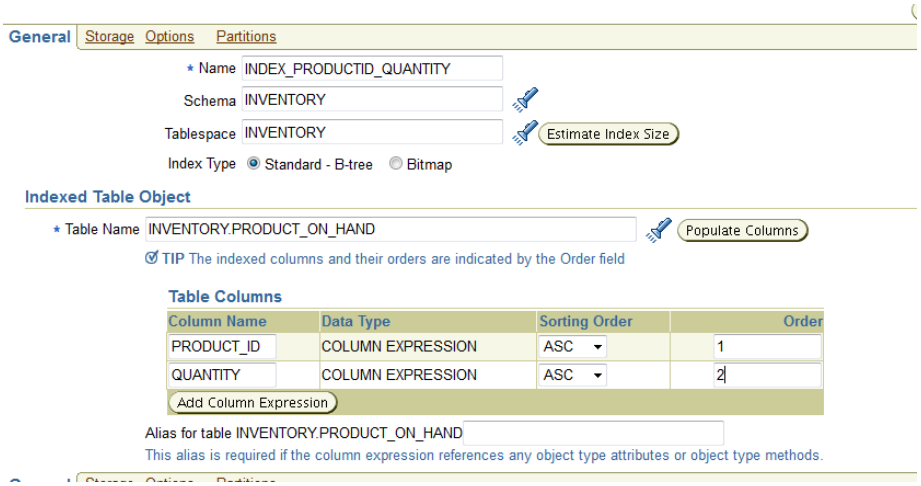



-When you click OK to create the index, you switch to a list of indexes for the INVENTORY schema. Why are there four indexes when you've created only one?

En plus de l'index que je viens de créer, trois indexes ont étés créés lorsque j'ai rajouté les contraintes "clé primaire" et "clé étrangère". Elles portent le nom que j'avais indiqué lors de la création des clés, sauf "SYS_C005320", qui concerne la clé primaire sur la table PRODUCT_ON_HAND, dù à un oubli d'avoir précisé un nom.

**5)**

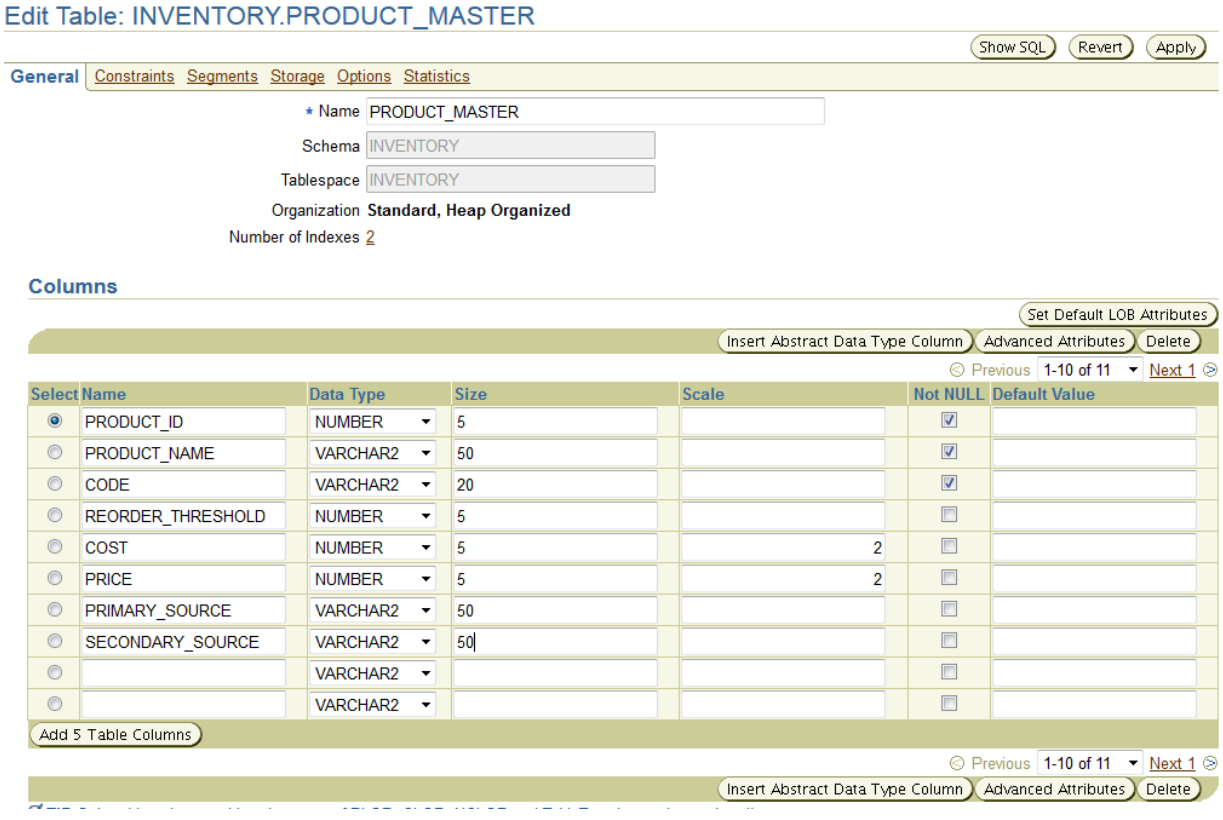
-Create an index on the PRODUCT_ID and QUANTITY columns of the PRODUCT_ON_HAND table.




**6)**

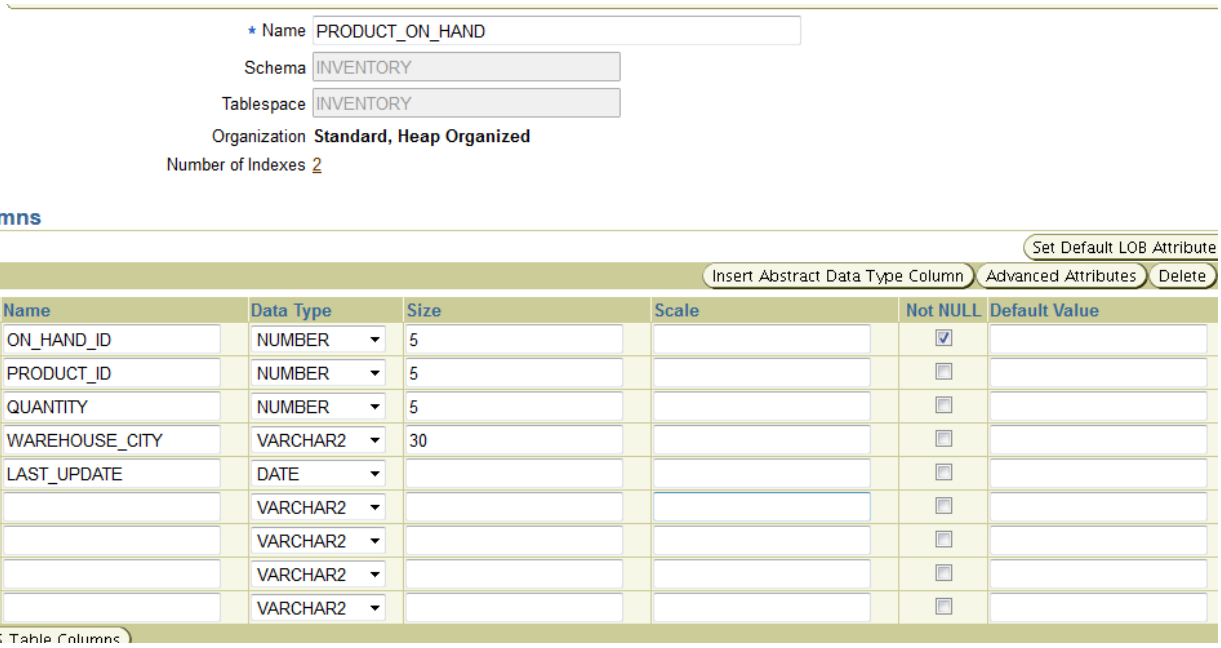
-You receive an update of the inventory application that requires you add two columns to the PRODUCT_MASTER table:

PRIMARY_SOURCE of datatype VARCHAR2(50)
SECONDARY_SOURCE, VARCHAR2(50).



**7)**

-The inventory application also requires you add the column LAST_UPDATE of datatype DATE in PRODUCT_ON_HAND table.



* Name PRODUCT_ON_HAND
Schema INVENTORY
Tablespace INVENTORY
Organization **Standard, Heap Organized**
Number of Indexes 2

**Columns**

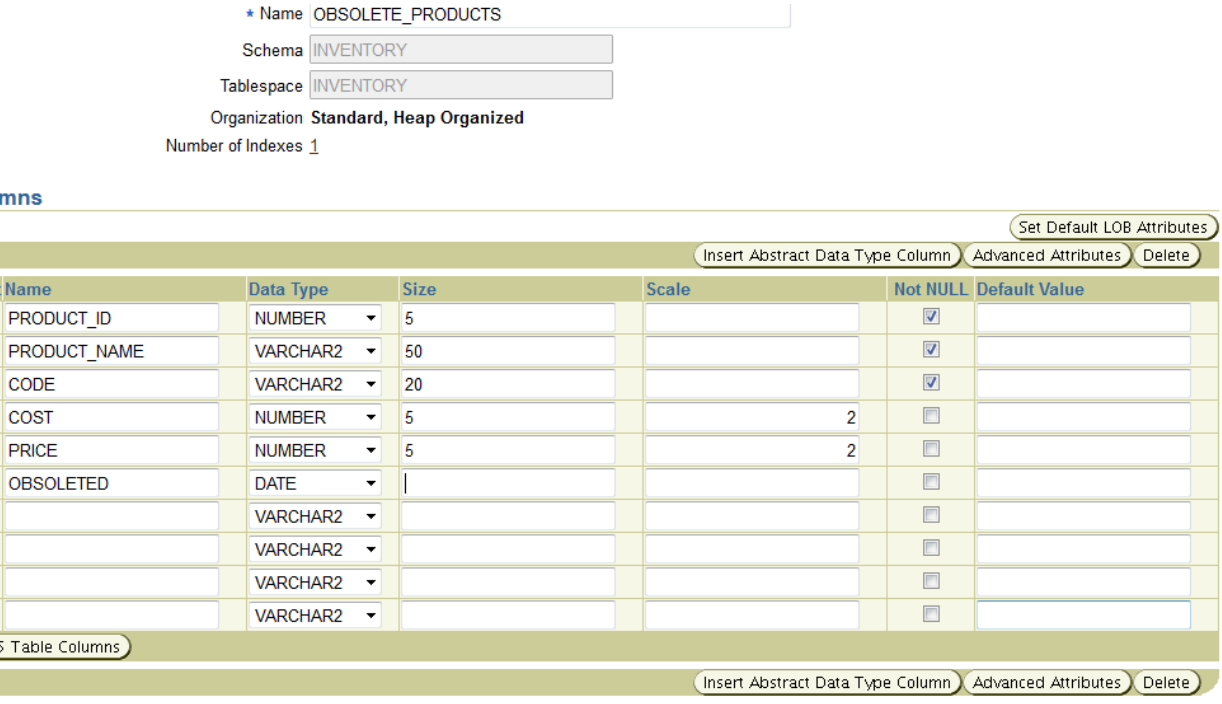
Set Default LOB Attribute | Insert Abstract Data Type Column | Advanced Attributes | Delete

| Select | Name | Data Type | Size | Scale | Not NULL | Default Value |
|---|---|---|---|---|---|---|
| ◉ | ON_HAND_ID | NUMBER | 5 | | ☑ | |
| ○ | PRODUCT_ID | NUMBER | 5 | | ☐ | |
| ○ | QUANTITY | NUMBER | 5 | | ☐ | |
| ○ | WAREHOUSE_CITY | VARCHAR2 | 30 | | ☐ | |
| ○ | LAST_UPDATE | DATE | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |

Add 5 Table Columns

**8)**

-Add a column named OBSOLETED of datatype DATE to the OBSOLETE_PRODUCTS table.



* Name OBSOLETE_PRODUCTS
Schema INVENTORY
Tablespace INVENTORY
Organization **Standard, Heap Organized**
Number of Indexes 1

**Columns**

Set Default LOB Attributes | Insert Abstract Data Type Column | Advanced Attributes | Delete

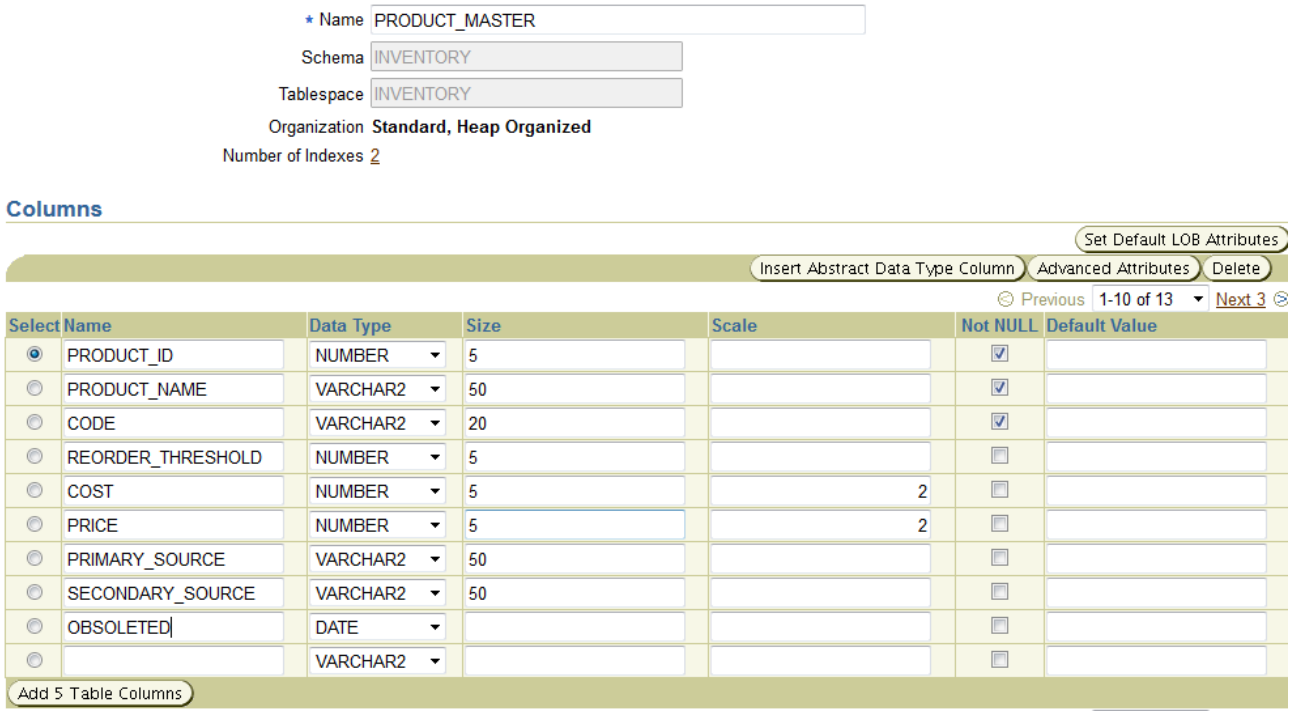
| Select | Name | Data Type | Size | Scale | Not NULL | Default Value |
|---|---|---|---|---|---|---|
| ◉ | PRODUCT_ID | NUMBER | 5 | | ☑ | |
| ○ | PRODUCT_NAME | VARCHAR2 | 50 | | ☑ | |
| ○ | CODE | VARCHAR2 | 20 | | ☑ | |
| ○ | COST | NUMBER | 5 | 2 | ☐ | |
| ○ | PRICE | NUMBER | 5 | 2 | ☐ | |
| ○ | OBSOLETED | DATE | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |
| ○ | | VARCHAR2 | | | ☐ | |

Add 5 Table Columns

Insert Abstract Data Type Column | Advanced Attributes | Delete

**9)**

-You receive another update for the inventory application. This update instructs you to drop the OBSOLETE_PRODUCTS table and add a column OBSOLETED to the PRODUCT_MASTER table.



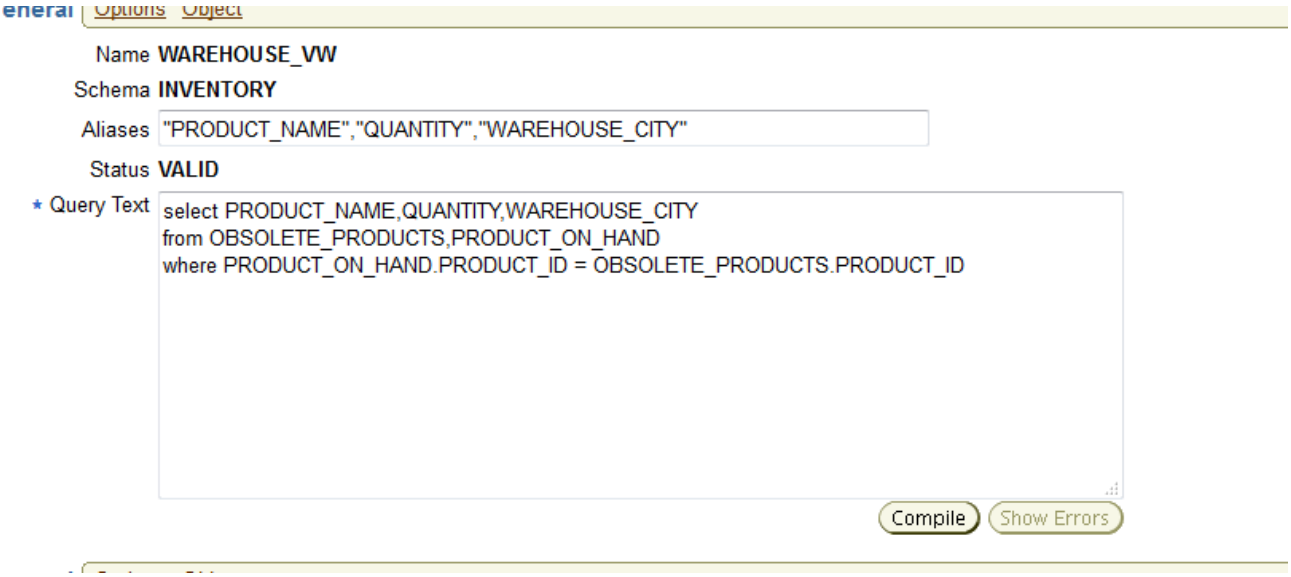

**10)**

-Then, you have to create a view named WAREHOUSE_VW that shows (in order):

The name of the product (product_name)

The amount of the product on hand (quantity)

The warehouse city name (warehouse_city)



## 6. *Manipulating Database Data*

### *6.1) data pump*

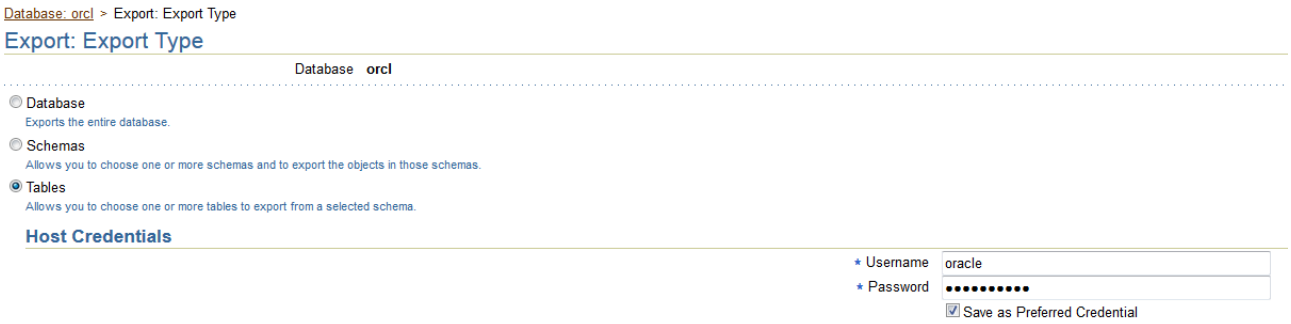
**1/2/3)**

-Create a directory to be used by Data Pump.
-Using SQL*Plus, log on to your instance as user SYSTEM and create a table to be used for testing Data Pump.
-Still within SQL*Plus, create the Oracle directory to be used by Data Pump and grant all users permission to read and write to the directory.

```
[oracle@vmware home]$ mkdir /home/oracle/tp_dtp
[oracle@vmware home]$ sqlplus

SQL*Plus: Release 10.1.0.3.0 - Production on Tue May 29 19:41:40 2018

Copyright (c) 1982, 2004, Oracle.  All rights reserved.

Enter user-name: system
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> CREATE TABLE dtp_test AS SELECT * FROM all_users:^[[D^H
  2
SQL> CREATE TABLE dtp_test AS SELECT * FROM all_users;

Table created.

Elapsed: 00:00:00.07
SQL> SELECT COUNT(*) FROM dtp_test;

  COUNT(*)
----------
        33

Elapsed: 00:00:00.00
SQL> CREATE DIRECTORY tp_dtp AS 'home/or^[[D
  2
SQL> CREATE DIRECTORY tp_dtp AS '/home/oracle/tp_dtp';

Directory created.

Elapsed: 00:00:00.02
SQL> GRANT read, write ON DIRECTORY tp_dtp TO PUBLIC;

Grant succeeded.

Elapsed: 00:00:00.02
SQL> exit
Disconnected from Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
[oracle@vmware home]$
```

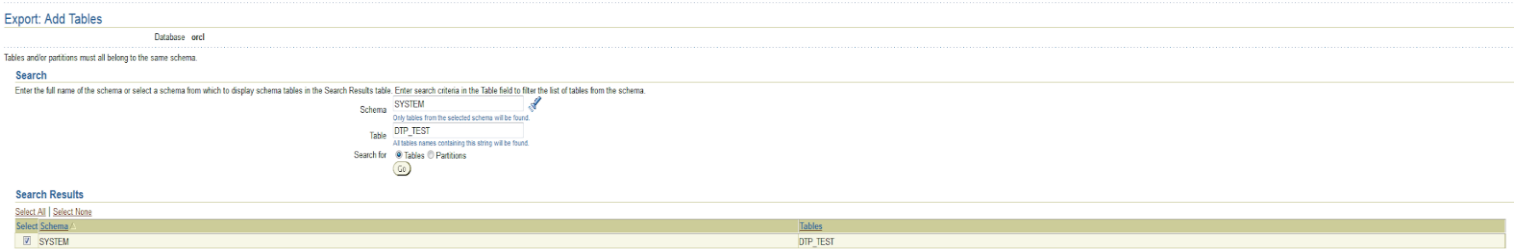
**7)**

-Select Tables and enter an operating system username and password with read/write permissions on the directory specified in step 3.



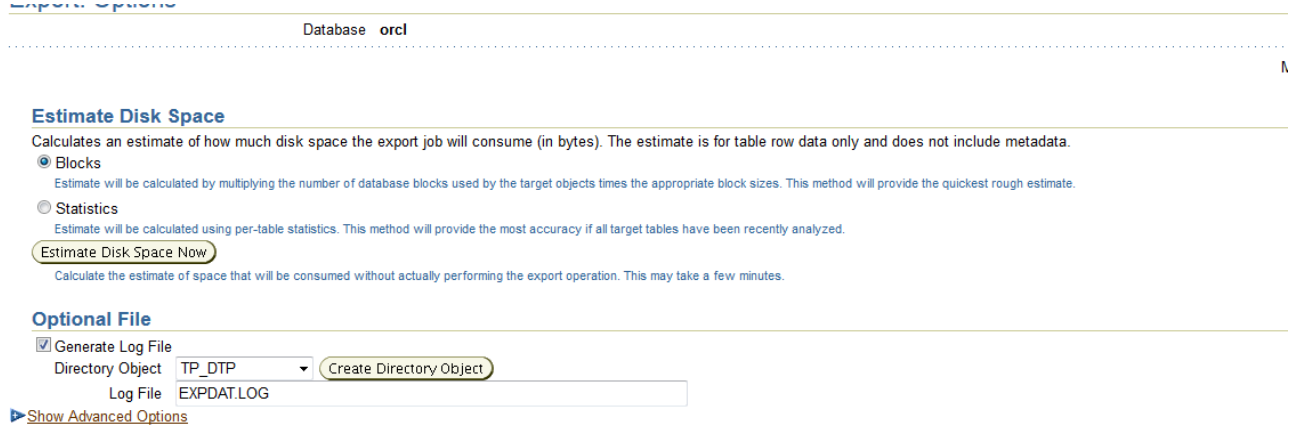

**8)**

-On the next screen, click Add, and specify SYSTEM as the schema and dtp_test as the table.



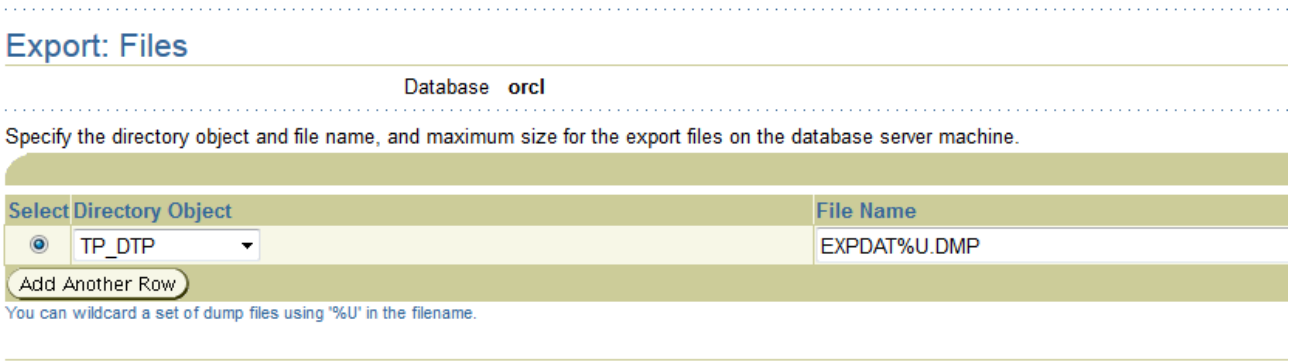

**9)**

-Then, choose the directory DTP_DIR as the location for the logfile.




**10)**

-Choose dtp_dir as the location.




**13)**

-From an operating system prompt, navigate to the directory specified in step 3. There will be a file EXPDAT01.DMP, which is the export dump file, and a lagfile, EXPDAT.LOG. Examine the logfile to check that the job did complete successfully.

```
[oracle@vmware home]$ cd oracle/tp_dtp/
[oracle@vmware tp_dtp]$ ls
EXPDAT01.DMP  EXPDAT.LOG
[oracle@vmware tp_dtp]$ cat EXPDAT.LOG
Job "SYSTEM"."EXPORT000003" stopped by user request at 22:35
Starting "SYSTEM"."EXPORT000003":
Job EXPORT000003 has been reopened at Tuesday, 29 May, 2018 22:35
Restarting "SYSTEM"."EXPORT000003":
Estimate in progress using BLOCKS method...
Processing object type TABLE_EXPORT/TABLE/TBL_TABLE_DATA/TABLE/TABLE_DATA
   estimated "SYSTEM"."DTP_TEST"                            64 KB
Total estimation using BLOCKS method: 64 KB
Processing object type TABLE_EXPORT/TABLE/TABLE
   . exported "SYSTEM"."DTP_TEST"                       6.226 KB      33 rows
Master table "SYSTEM"."EXPORT000003" successfully loaded/unloaded
******************************************************************************
Dump file set for SYSTEM.EXPORT000003 is:
  /home/oracle/tp_dtp/EXPDAT01.DMP
Job "SYSTEM"."EXPORT000003" successfully completed at 22:36
[oracle@vmware tp_dtp]$
```

## *6.2) SQL loader*

**1)**

-Use the control file lab_12_a.ctl to load data from the text file lab_12_a.dat into the PRODUCT_MASTER table.

```
load data
infile 'lab_12_a.dat' "str '\n'"
append
into table INVENTORY.PRODUCT_MASTER
fields terminated by ','
(PRODUCT_ID,PRODUCT_NAME,CODE)
```

```
GNU nano 1.2.1                                    File: lab_12_a.dat

,pomme,01
,poison,02
,mirroir,03
,nain de jardin,04
```

```
[oracle@vmware orcl]$ sqlldr userid=\'sys/oracle AS SYSDBA\' control=lab_12_a.ctl

SQL*Loader: Release 10.1.0.3.0 - Production on Tue May 29 23:04:40 2018

Copyright (c) 1982, 2004, Oracle.  All rights reserved.

Commit point reached - logical record count 5
[oracle@vmware orcl]$
```

**2)**

- Use the control file

```
SQL> select * from inventory.product_master;

PRODUCT_ID PRODUCT_NAME
---------- --------------------------------------------------
CODE                 REORDER_THRESHOLD       COST      PRICE
-------------------- ---------------- ---------- ----------
PRIMARY_SOURCE
--------------------------------------------------
SECONDARY_SOURCE                                   OBSOLETED
-------------------------------------------------- ---------
         1 pomme
01



         2 poison
02



         3 mirroir
03



         4 nain de jardin
04
```

lab_12_f.ctl to load data from the text file lab_12_f.dat into the PRODUCT_ON_HAND table.

```
 GNU nano 1.2.1                                          File: lab_12_f.ctl

load data
infile 'lab_12_f.dat' "str '\n'"
append
into table INVENTORY.PRODUCT_ON_HAND
fields terminated by ','
(ON_HAND_ID,PRODUCT_ID)
```

```
 GNU nano 1.2.1                                          File: lab_12_f.dat

1,01
2,02
3,03
4,04
```

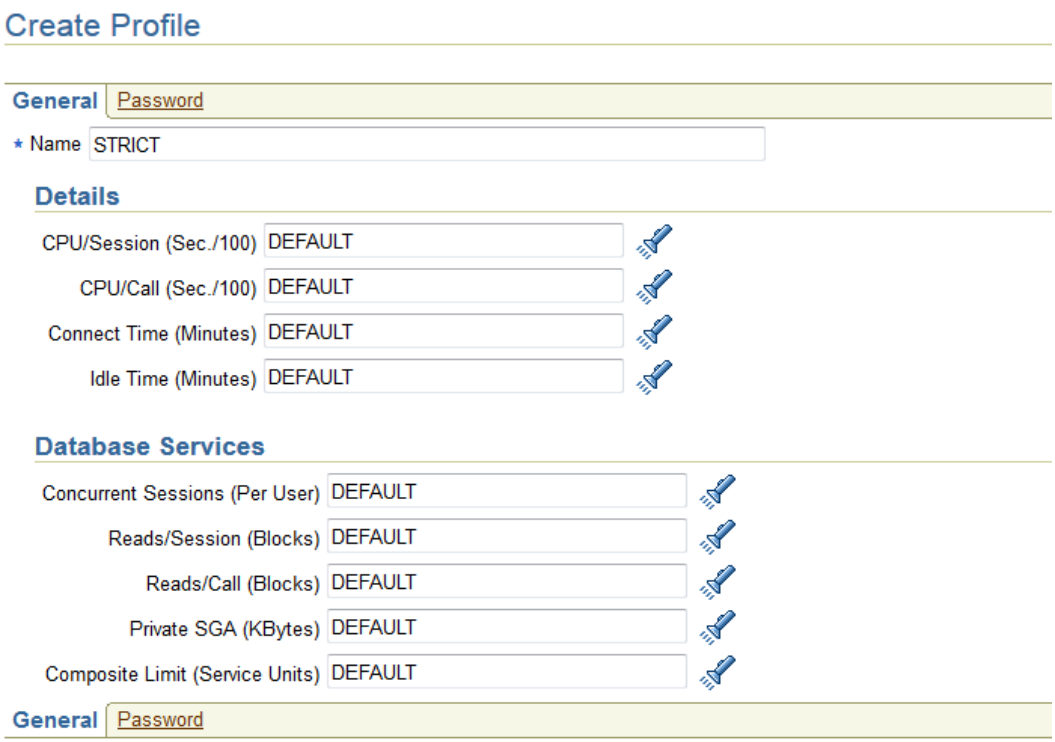
```
[oracle@vmware orcl]$ sqlldr userid=\'sys/oracle AS SYSDBA\' control=lab_1

SQL*Loader: Release 10.1.0.3.0 - Production on Wed May 30 00:10:42 2018

Copyright (c) 1982, 2004, Oracle.  All rights reserved.

Commit point reached - logical record count 5
```

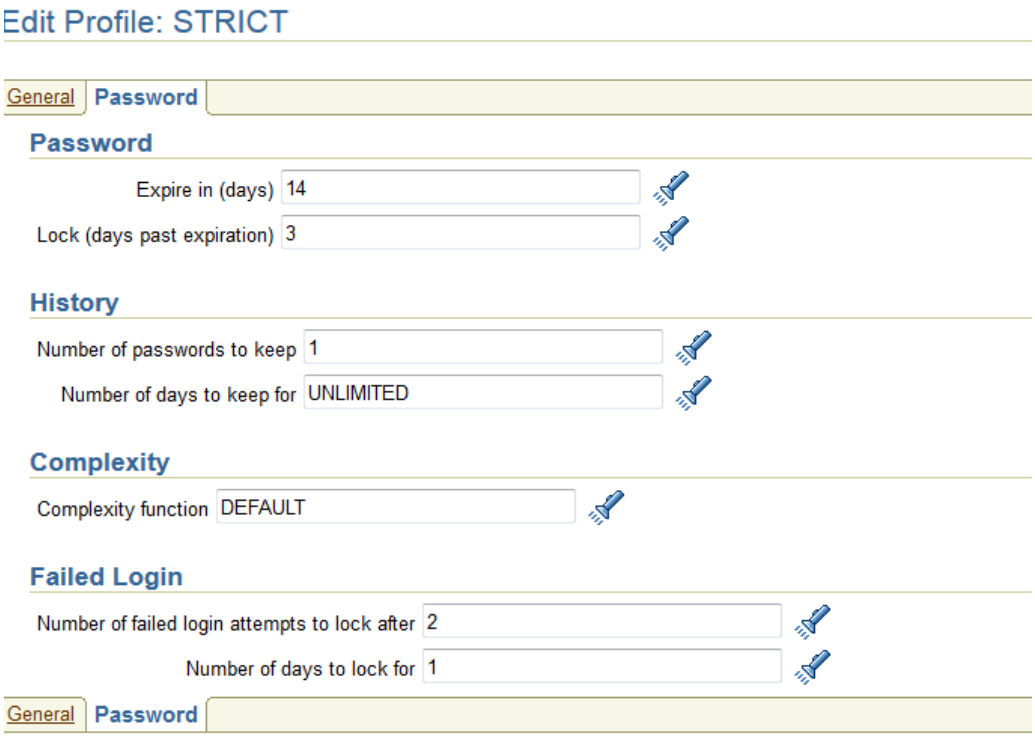
```
Elapsed: 00:00:00.00
SQL> select * from inventory.product_on_hand;

ON_HAND_ID PRODUCT_ID   QUANTITY WAREHOUSE_CITY                 LAST_UPDA
---------- ---------- ---------- ------------------------------ ---------
         1          1
         2          2
         3          3
         4          4

Elapsed: 00:00:00.01
```

## 7. *Creating and Using Pasword Profiles*

**3/4)**

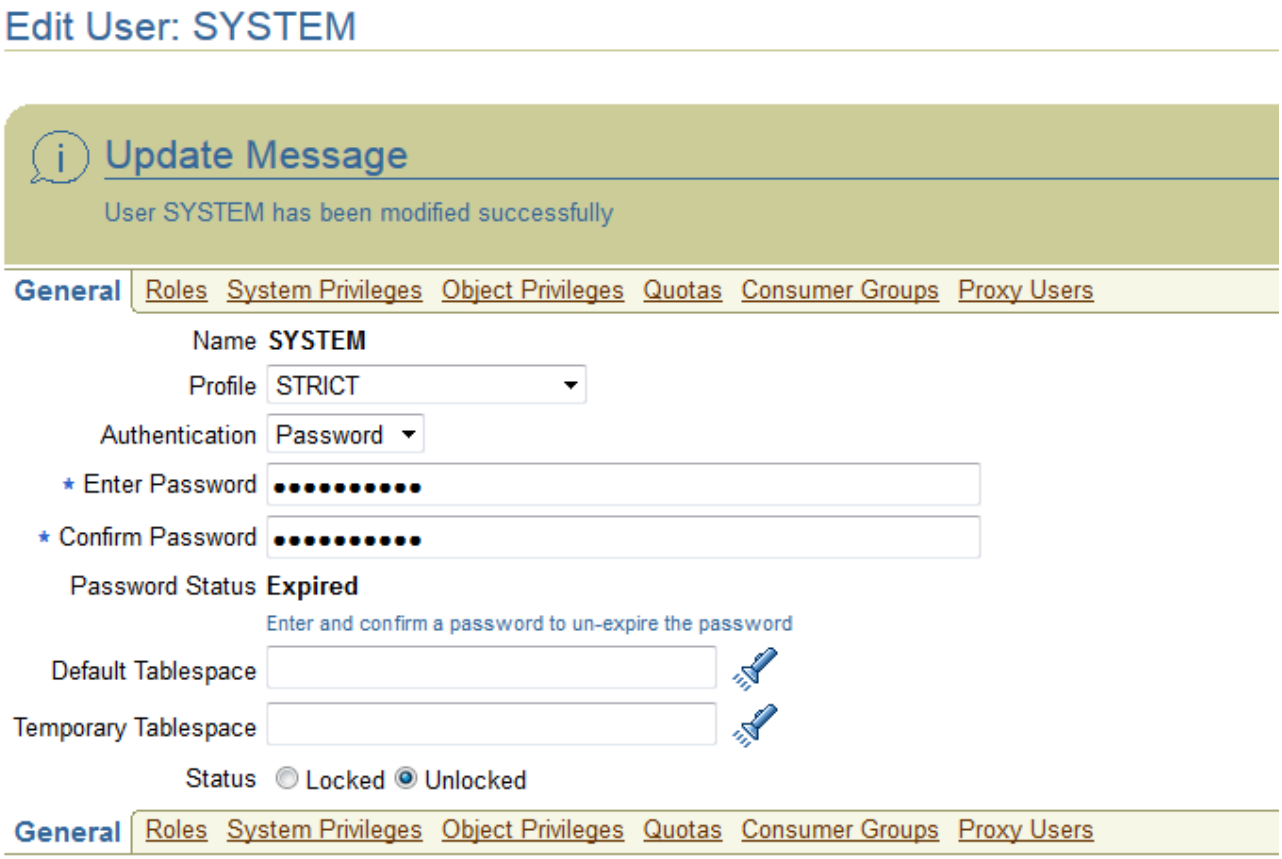
-Click Create to reach the Create Profile window, and enter STRICT as the profile name. Take the Password link to reach the password controls window.
-Set limits for your STRICT profile. Users assigned to this profile will have to change their passwords after two weeks, and they will have three days to do so. A password can only ever be used once, and after two failed login attempts the account will be locked, but only for one minute.



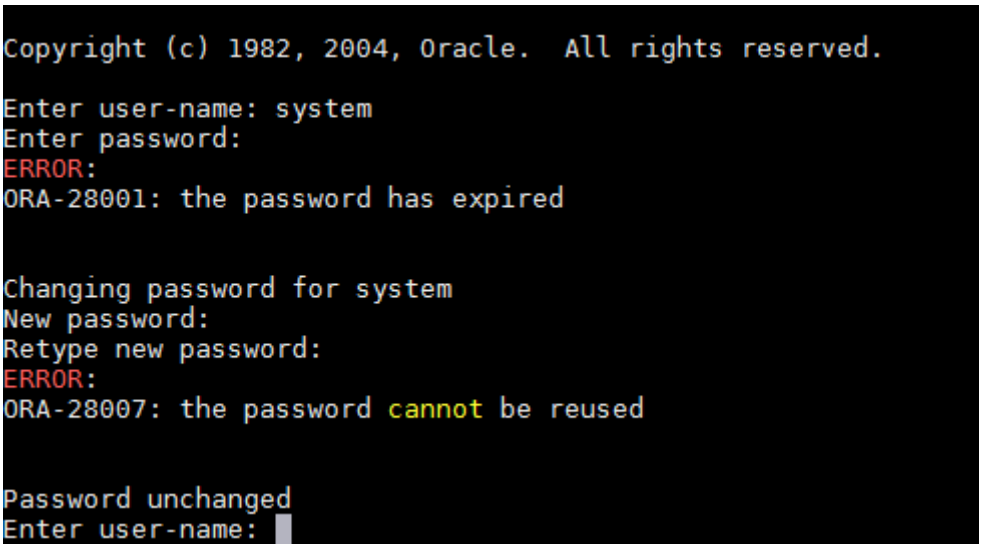

**9)**

-In the Edit User: SYSTEM window, select the STRICT profile and expire the password.




**13)**

-Attempt to change the password to the value it is already.



**15)**

-Attempt to connect three times with the wrong password. At the third attempt, you will be told that the account is locked. Wait at least one minute, and then connect with the correct password.

```
oracle@vmware orcl]$ sqlplus

QL*Plus: Release 10.1.0.3.0 - Production on Wed May 30 01:23:24 2018

opyright (c) 1982, 2004, Oracle.  All rights reserved.

nter user-name: system
nter password:
RROR:
RA-01017: invalid username/password; logon denied


nter user-name: system
nter password:
RROR:
RA-01017: invalid username/password; logon denied


nter user-name: system
nter password:
RROR:
RA-28000: the account is locked


P2-0157: unable to CONNECT to ORACLE after 3 attempts, exiting SQL*Plus
oracle@vmware orcl]$ sqlplus
```

**16)**

-Tidy up by assigning **SYSTEM** back to the default profile and dropping the STRICT profile.

```
Connected to:
Oracle Database 10g Enterprise Edition Release 10.
With the Partitioning, OLAP and Data Mining option

SQL> ALTER USER systel^H
  2
SQL> a^H
  1* ALTER USER systel
SQL>
SQL> ALTER USER system PROFILE default;

User altered.

Elapsed: 00:00:00.03
SQL> DROP Profile strict;

Profile dropped.

Elapsed: 00:00:00.08
```

## 8. *Enabling Auditing*

**2)**

-Set the **AUDIT_TRAIL** instance parameter to enable auditing to the data dictionary. As this is a static parameter, you must use the **SCOPE** clause and restart the instance.

```
Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL>
SQL>
SQL> ALTER SYSTEM SET audit_trail=db SCOPE=spfile;

System altered.

Elapsed: 00:00:00.01
SQL> STARTUP FORCE;
ORACLE instance started.

Total System Global Area  167772160 bytes
Fixed Size                   778212 bytes
Variable Size              61874204 bytes
Database Buffers          104857600 bytes
Redo Buffers                 262144 bytes
Database mounted.
Database opened.
SQL>
```

**4)**

-Create a table and insert some rows.

```
Enter user-name: system
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> CREATE TABLE audit_test (name VARCHAR2(10), salary NUMBER);

Table created.

Elapsed: 00:00:00.09
SQL> INSERT INTO audit_test VALUES ('McGraw',100);

1 row created.

Elapsed: 00:00:00.01
SQL> INSERT INTO audit_test VALUES ('Hill',200);
```

**5)**

-Enable database auditing of access to the table.

```
Elapsed: 00:00:00.01
SQL> AUDIT select,update ON system.audit_test;

Audit succeeded.
```

**6)**

-Execute some statements against the table.

```
Elapsed: 00:00:00.04
SQL> SELECT * FROM audit_test;

NAME           SALARY
---------- ----------
McGraw            100
Hill              200

Elapsed: 00:00:00.03
SQL> Upda^H^H^H
SP2-0042: unknown command "U" - rest of line ignored.
SQL> UPDATE audit_test SET salary = 50
  2  WHERE name='McGraw';

1 row updated.
```

**7)**

-Query the DBA_AUDIT_TRAIL view to see the results of the auditing.

```
Elapsed: 00:00:00.03
SQL> SELECT username,userhost,os_username,ses_actions,obj_name FROM dba_audit_trail;

USERNAME
------------------------------
USERHOST
--------------------------------------------------------------------------------
OS_USERNAME
--------------------------------------------------------------------------------
SES_ACTIONS
-------------------
OBJ_NAME
--------------------------------------------------------------------------------
SYSTEM
vmware.labo-oracle.com
oracle
---------SS-----
AUDIT_TEST


Elapsed: 00:00:00.05
```

**8)**

-Create an FGA policy to capture all SELECTs against the AUDIT_TEST table that read the SALARY column, if the salary retrieved is greater than 100.

```
Elapsed: 00:00:00.05
SQL> EXEC dbms_fga.add_policy ( object_schema =>'system', object_name => 'audit_test',policy_name => 'high_sal',audit_condition => 'salary > 100',audit_colu
mn => 'salary',statement_types => 'select');

PL/SQL procedure successfully completed.
```

**9)**

-Run some queries against the table.

```
Elapsed: 00:00:00.08
SQL>
SQL> SELECT * FROM audit_test;

NAME           SALARY
---------- ----------
McGraw             50
Hill              200

Elapsed: 00:00:00.05
SQL> SELECT salary FROM audit_test WHERE name='Hill';

    SALARY
----------
       200

Elapsed: 00:00:00.01
SQL> SELECT salary FROM audit_test WHERE name='McGraw';

    SALARY
----------
        50

Elapsed: 00:00:00.01
SQL> SELECT name FROM audit_test;

NAME
----------
McGraw
Hill

Elapsed: 00:00:00.01
SQL>
```

**10)**

-Query the fine-grained audit trail.

```
SQL> select os_user,db_user,sql_text from dba_fga_audit_trail;

OS_USER
--------------------------------------------------------------------------------
DB_USER
------------------------------
SQL_TEXT
--------------------------------------------------------------------------------
oracle
SYSTEM
SELECT * FROM audit_test

oracle
SYSTEM
SELECT salary FROM audit_test WHERE name='Hill'
```

**11)**

-Tidy up by canceling the database auditing, dropping the FGA policy, and dropping the table.

```
SQL> NOAUDIT select, update ON system.audit_test;

Noaudit succeeded.

Elapsed: 00:00:00.04
SQL> EXEC dbms_fga.drop_policy ( objeck_name
BEGIN dbms_fga.drop_policy ( objeck_name; END;

                                        *
ERROR at line 1:
ORA-06550: line 1, column 41:
PLS-00103: Encountered the symbol ";" when expecting one of the following:
. ( ) , * @ % & | = - + < / > at in is mod remainder not
range rem => .. <an exponent (**)> <> or != or ~= >= <= <>
and or like as between from using || multiset member
SUBMULTISET_
The symbol ")" was substituted for ";" to continue.


Elapsed: 00:00:00.01
SQL> EXEC dbms_fga.drop_policy ( object_name => 'audit_test', policy_name => 'high_sal');

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.03
SQL> d^H
SP2-0042: unknown command "" - rest of line ignored.
SQL> DROP TABLE audit_test;

Table dropped.

Elapsed: 00:00:00.31
SQL>
```

## 9. *monitoring and Management*

### *9.1)Managing Database Performance*

#### *9.1.1)Repairing Invalid Objects*

**2)**

-Create a user TESTUSER to be used for this exercise; grant him the DBA privilege.

```
SQL> CREATE USER TESTUSER IDENTIFIED BY oracle;

User created.

Elapsed: 00:00:00.05
SQL> GRANT dba To^H
  2
SQL> GRAND
SP2-0042: unknown command "GRAND" - rest of line ignored.
SQL> GRANT DBA TO testuser IDENTIFIED BY oracle;

Grant succeeded.

Elapsed: 00:00:00.06
SQL>
```

**3)**

-Connect as TESTUSER, and create some objects.

```
SQL> CREATE TABLE testtab(n1 NUMBER,d1 DATE);

Table created.

Elapsed: 00:00:00.08
SQL> INSERT INTO testtab v^H
  2  ;
INSERT INTO testtab v
                    *
ERROR at line 1:
ORA-00911: invalid character


Elapsed: 00:00:00.01
SQL> INSERT INTO testtab VALUES (1,SYSDATE);

1 row created.

Elapsed: 00:00:00.01
SQL> CREATE OR REPLACE VIEW v1 AS SELECT d1 FROM testtab;

View created.

Elapsed: 00:00:00.03
SQL> CREATE OR REPLACE PROCEDURE p1 AS cnt NUMBER;
  2  BEGIN
  3  SELECT COUNT(*) INTO cnt FROM testtab;
  4  END;
  5  /

Procedure created.

Elapsed: 00:00:00.10
```

**4)**

-Confirm the status of the objects.

```
SQL> select object_name,object_type,status FROM user_objects;

OBJECT_NAME
--------------------------------------------------------------------------------
OBJECT_TYPE         STATUS
------------------- -------
V1
VIEW                VALID

TESTTAB
TABLE               VALID

P1
PROCEDURE           VALID


Elapsed: 00:00:00.05
```

**5)**

-Perform a DDL command on the table.

```
SQL> ALTER TABLE testtab DROP COLUMN d1;

Table altered.

Elapsed: 00:00:00.13
```

**6)**

-Re-run the query from step 4. Note that both the procedure and the view are now INVALID.

```
Elapsed: 00:00:00.05
SQL> ALTER TABLE testtab DROP COLUMN d1;

Table altered.

Elapsed: 00:00:00.11
SQL> SELECT object_name,object_type,status FROM user_objects;

OBJECT_NAME
--------------------------------------------------------------------------------
OBJECT_TYPE         STATUS
------------------- -------
V1
VIEW                INVALID

TESTTAB
TABLE               VALID

P1
PROCEDURE           INVALID


Elapsed: 00:00:00.05
SQL>
```

**7)**

-Recompile the procedure.

```
Elapsed: 00:00:00.05
SQL> ALTER PROCEDURE p1 COMPILE;

Procedure altered.

Elapsed: 00:00:00.12
```

**8)**

-Re-compile the view.

```
SQL> ALTER VIEW v1 COMPILE;

Warning: View altered with compilation errors.

Elapsed: 00:00:00.09
```

**9)**

-To diagnose the problem, query the DBA_DEPENDENCIES view.

```
REFERENCED_NAME
--------------------------------------------------------------------
REFERENCED_OWNER               REFERENCED_TYPE
------------------------------ -----------------
TESTTAB
TESTUSER                       TABLE

D1
TESTUSER                       NON-EXISTENT

D1
PUBLIC                         NON-EXISTENT


Elapsed: 00:00:00.06
SQL>
```

**10)**

-To pinpoint the exact problem, retrieve the code on which the view is based.

```
SQL> SELECT text FROM user_views WHERE view_name = 'V1';

TEXT
--------------------------------------------------------------------
SELECT d1 FROM testtab
```

**11)**

-To fix the problem, add the column back to the table and recompile.

```
SQL> ALTER TABLE testtab ADD (d1 DATE);

Table altered.

Elapsed: 00:00:00.08
SQL> ALTER VIEW v1 COMPL
  2
SQL> ALTER VIEW v1 COMPILR
  2
SQL> ALTER VIEW v1 COMPILE;

View altered.

Elapsed: 00:00:00.08
SQL>
```

**12)**

-Confirm that all the objects are now valid by re-running the query from step 4.

```
SQL> SELECT object_name,object_type,status FROM user_objects;

OBJECT_NAME
--------------------------------------------------------------------------------
OBJECT_TYPE         STATUS
------------------- -------
V1
VIEW                VALID

TESTTAB
TABLE               VALID

P1
PROCEDURE           VALID
```

**13)**

-Tidy up by dropping view and procedure.

```
SQL> DROP VIEW v1;

View dropped.

Elapsed: 00:00:00.17
SQL> DROP PROCEDURE p1;

Procedure dropped.

Elapsed: 00:00:00.08
SQL>
```

*9.1.2)Repairing Unusable Indexes*

**1)**

-In your SQL*Plus session, connect as TESTUSER and create two indexes.

```
SQL> CREATE INDEX d1_idx ON testtab(d1);

Index created.

Elapsed: 00:00:00.05
SQL> CREATE INDEX n1_idx ON testtab(n1);

Index created.
```

**2)**

-Confirm the index creation and status. Both will be VALID.

```
SQL> SELECT index_name,status FROM user_indexes;

INDEX_NAME                     STATUS
------------------------------ --------
D1_IDX                         VALID
N1_IDX                         VALID
```

**3)**

-Move the table.

```
SQL> alter table testtab MOVE;

Table altered.
```

**4)**

-Run the query from step 2 again. The move of the table, which changed any rowids, will have rendred the indexes unusable.

```
SQL> select index_name,status from user_indexes;

INDEX_NAME                     STATUS
------------------------------ --------
D1_IDX                         UNUSABLE
N1_IDX                         UNUSABLE
```

**5)**

-Rebuild one index, using the NOLOGGING and ONLINE options.

```
SQL> ALTER INDEX n1_idx REBUILD ONLINE NOLOGGING;

Index altered.
```

**8)**

-In the Search section of the Indexes window, enter TESTUSER as the Schema, and click Go. This will show the two indexes on the TESTTAB table, one of which, D1_IDX, is still unusable.

**Search**

Select an object type and optionally enter a schema name and an object name to filter the data that is displayed in your results set.

| Object Type | Search By | Schema | Object Name | |
|---|---|---|---|---|
| Index | Table Name | TESTUSER | | Go |

To run an exact match search or to run a case sensitive search, double quote the search criteria. The wildcard (%) symbol can still be used in a double quoted search string.

**Results**

| Select | Table Owner | Table | Index Owner | Index | Index Type | Table Type |
|---|---|---|---|---|---|---|
| ◉ | TESTUSER | TESTTAB | TESTUSER | D1_IDX | NORMAL | TABLE |
| ○ | TESTUSER | TESTTAB | TESTUSER | N1_IDX | NORMAL | TABLE |

**9)**

-Select the radio button for the unusable index, select Reorganize in the Actions drop-down box, and click Go to launch the Reorganize Objects Wizard.

## Reorganize Objects: Objects

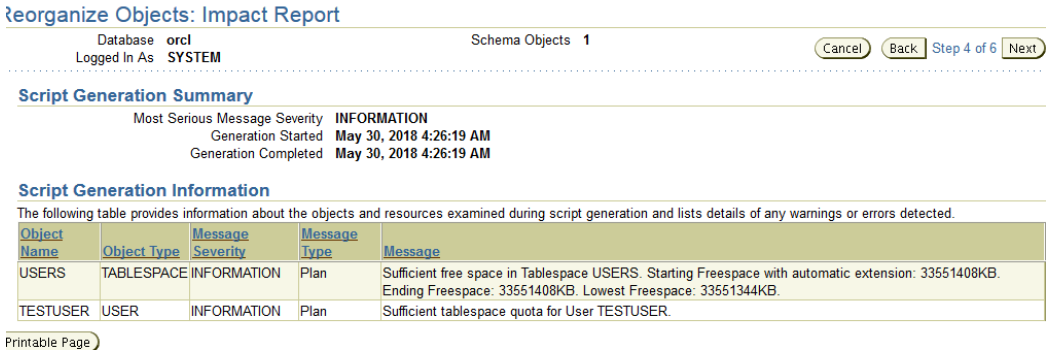
Database **orcl**
Logged In As **SYSTEM**

This table contains the schema objects to be reorganized. Click Add to add schema objects to the table.

| Select | Name | Type | Current Tablespace |
|---|---|---|---|
| ◉ | TESTUSER.D1_IDX | Index | USERS |

**10)**

-Click Next, leave all the options on default, and click Next again to generate the reorganization script and reach the Impact Report window. This should confirm that there is sufficient free space for the operation to proceed. Click Next to proceed.



Reorganize Objects: Impact Report

Database orcl
Logged In As SYSTEM

Schema Objects 1

Cancel | Back | Step 4 of 6 | Next

**Script Generation Summary**

Most Serious Message Severity INFORMATION
Generation Started May 30, 2018 4:26:19 AM
Generation Completed May 30, 2018 4:26:19 AM

**Script Generation Information**

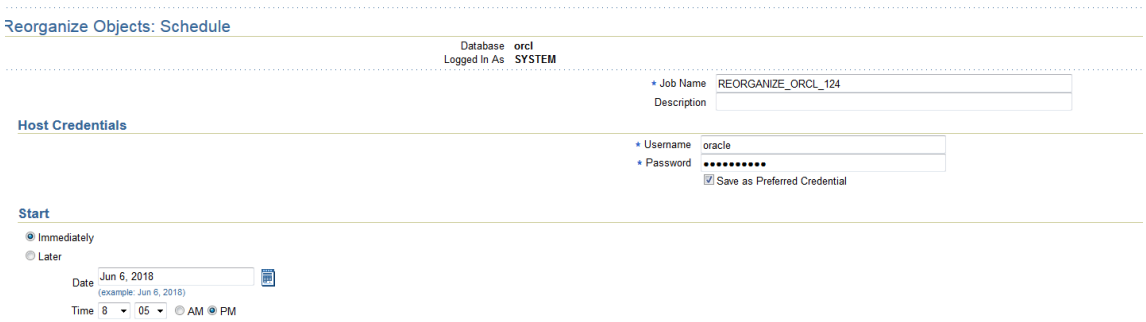
The following table provides information about the objects and resources examined during script generation and lists details of any warnings or errors detected.

| Object Name | Object Type | Message Severity | Message Type | Message |
|---|---|---|---|---|
| USERS | TABLESPACE | INFORMATION | Plan | Sufficient free space in Tablespace USERS. Starting Freespace with automatic extension: 33551408KB. Ending Freespace: 33551408KB. Lowest Freespace: 33551344KB. |
| TESTUSER | USER | INFORMATION | Plan | Sufficient tablespace quota for User TESTUSER. |

Printable Page

**11)**

-On the Reorganize Objects: Schedule window, leave everything on default to run the job immediately, and click Next to reach the Review window.



Reorganize Objects: Schedule

Database orcl
Logged In As SYSTEM

* Job Name REORGANIZE_ORCL_124
Description

**Host Credentials**

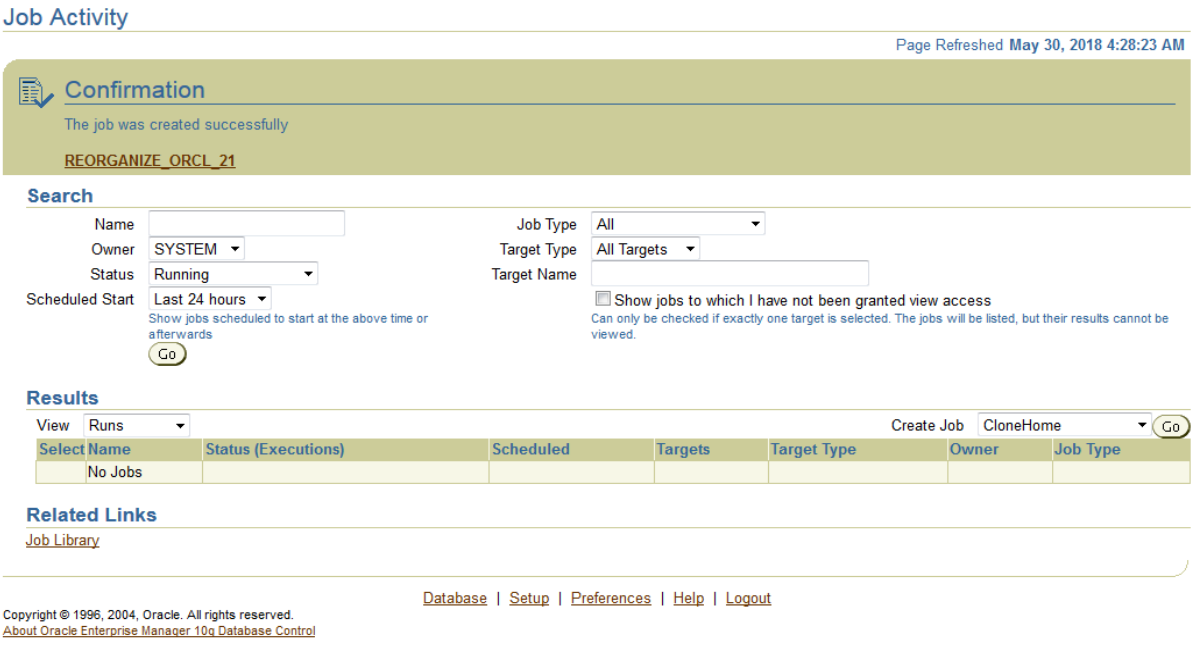
* Username oracle
* Password ••••••••••
Save as Preferred Credential

**Start**

Immediately
Later
Date Jun 6, 2018 (example: Jun 6, 2018)
Time 8 05 AM PM

**12)**

-In the Review window, click Submit Job to rebuild the index.



Job Activity

Page Refreshed May 30, 2018 4:28:23 AM

**Confirmation**

The job was created successfully

REORGANIZE_ORCL_21

**Search**

Name
Owner SYSTEM
Status Running
Scheduled Start Last 24 hours
Show jobs scheduled to start at the above time or afterwards
Go

Job Type All
Target Type All Targets
Target Name
Show jobs to which I have not been granted view access
Can only be checked if exactly one target is selected. The jobs will be listed, but their results cannot be viewed.

**Results**

View Runs
Create Job CloneHome Go

| Select | Name | Status (Executions) | Scheduled | Targets | Target Type | Owner | Job Type |
|---|---|---|---|---|---|---|---|
| | No Jobs | | | | | | |

**Related Links**

Job Library

Database | Setup | Preferences | Help | Logout

**13)**

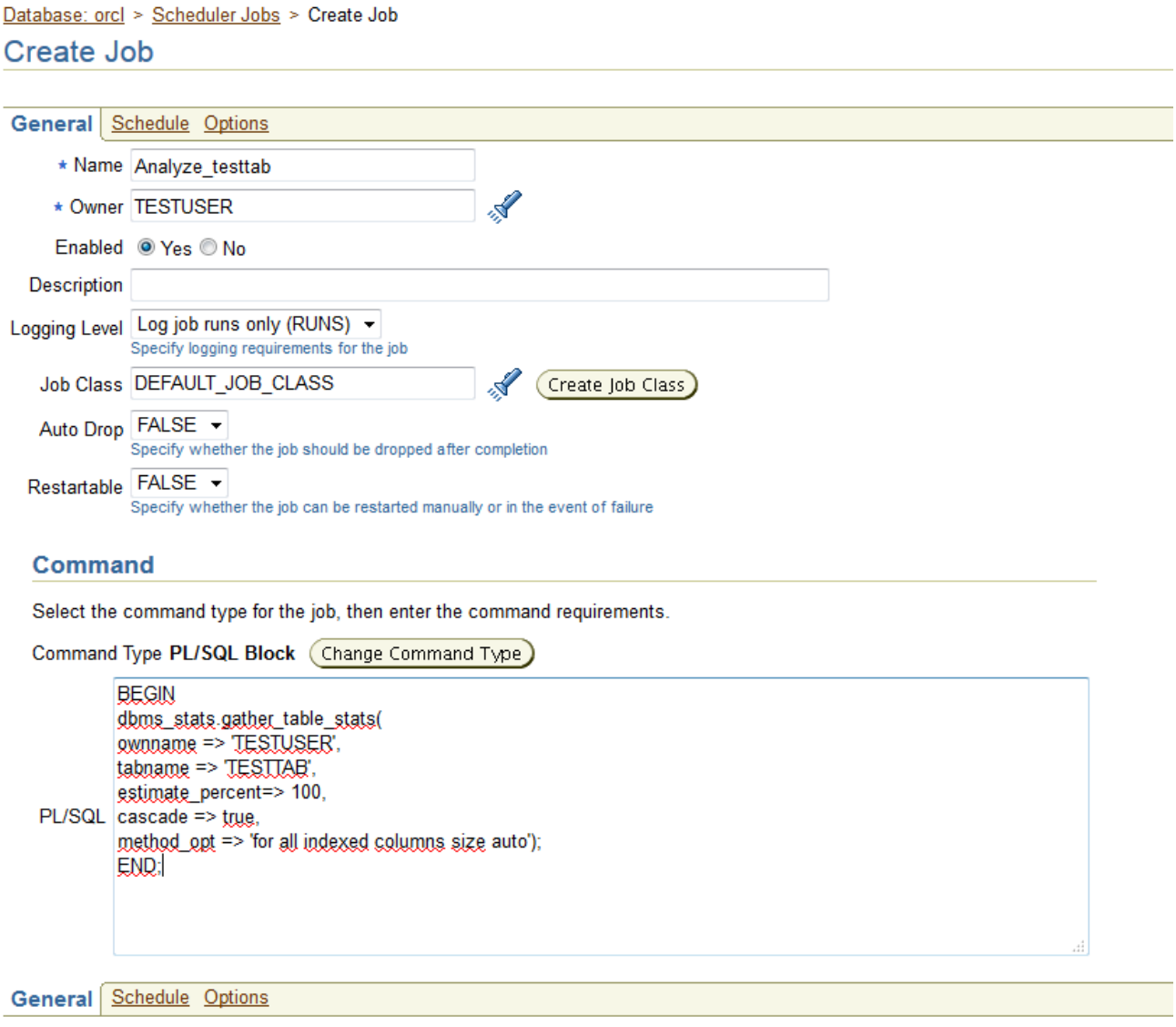
-In your SQL*Plus session, confirm that the index is now valid by running the query from step 2.

```
Elapsed: 00:00:00.01
SQL> SELECT index_name,status FROM user_indexes;

INDEX_NAME                     STATUS
------------------------------ --------
D1_IDX                         VALID
N1_IDX                         VALID

Elapsed: 00:00:00.08
```

*9.1.3)Automating Statistics Collection*

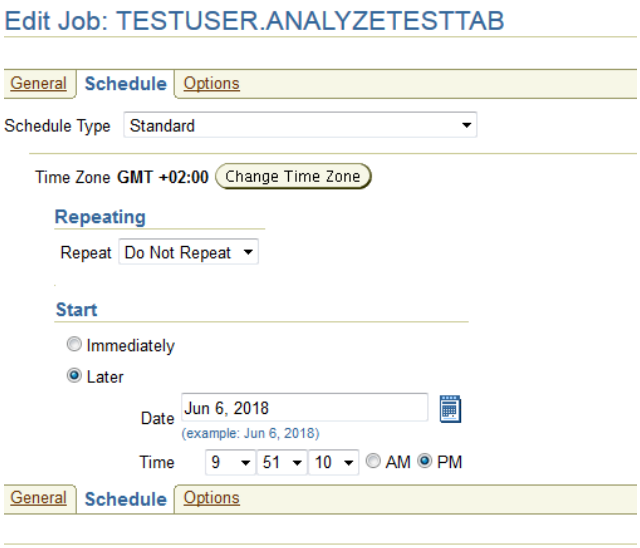
**3/4)**

-Click Create to reach the Create Job window. In the Credential section, enter the Name as Analyze testtab, and leave everything else on default.
-In the Command section, replace the sample code.



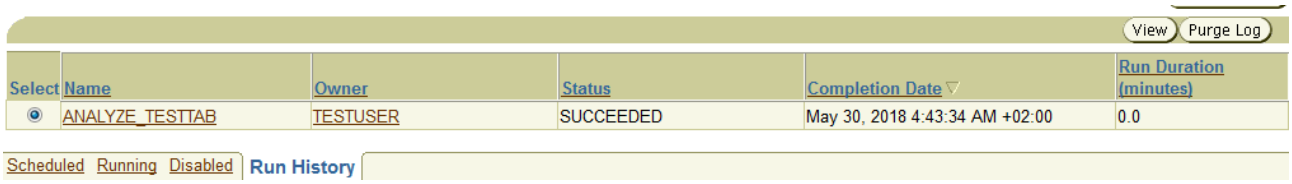
**5)**

-Take the Schedule link to reach the Schedule window. Leave everything on default, to run the job once only right away, and return to the Scheduler Jobs window.




**6)**

-Take the Run History link, and you will see that the job has succeeded.




**7)**

-In your SQL*Plus session, set your NLS_DATE_FORMAT session parameter to show the full time and confirm that statistics were indeed collected.

```
SQL> SELECT * FROM USER_SCHEDULER_JOB_LOG;

    LOG_ID
----------
LOG_DATE
---------------------------------------------------------------------------
OWNER                          JOB_NAME
------------------------------ ------------------------------
JOB_CLASS                      OPERATION
------------------------------ ------------------------------
STATUS                         USER_NAME
------------------------------ ------------------------------
CLIENT_ID
----------------------------------------------------------------
GLOBAL_UID
--------------------------------
ADDITIONAL_INFO
------------------------------------------------------------------------------
       165
30-MAY-18 04.43.34.131679 AM +02:00
TESTUSER                       ANALYZE_TESTTAB
DEFAULT_JOB_CLASS              RUN
SUCCEEDED




       166
30-MAY-18 04.43.34.133586 AM +02:00
TESTUSER                       ANALYZE_TESTTAB
DEFAULT_JOB_CLASS              SUCCEEDED


REASON="Run once job disabled"
```

**8)**

-Tidy up by connecting as user SYSTEM and dropping the TESTUSER schema.

```
Elapsed: 00:00:00.07
SQL> DROP USER TESTUSER CASCADE;

User dropped.

Elapsed: 00:00:02.61
SQL>
```

## *9.2) Monitoring Oracle*

### *9.2.1) Generating an ADDM Report*

**2)**

-Force the creation of an AWR snapshot.

```
SQL> EXEC dbms_workload_repository.create_snapshot;

PL/SQL procedure successfully completed.

Elapsed: 00:00:10.53
```

**3)**

-Simulate a workload by creating a table and running this anonymous PL/SQL block to generate some activity.

```
SQL> CREATE TABLE tmptab AS SELECT * FROM all_objects;

Table created.

Elapsed: 00:00:00.90
SQL> BEGIN
  2  FOR i IN 1..10 LOOP
  3  INSERT INTO tmptab
  4  SELECT * FROM all_objects;
  5  DELETE FROM tmptab;
  6  END LOOP;
  7  COMMIT;
  8  END;
  9  /


PL/SQL procedure successfully completed.
```

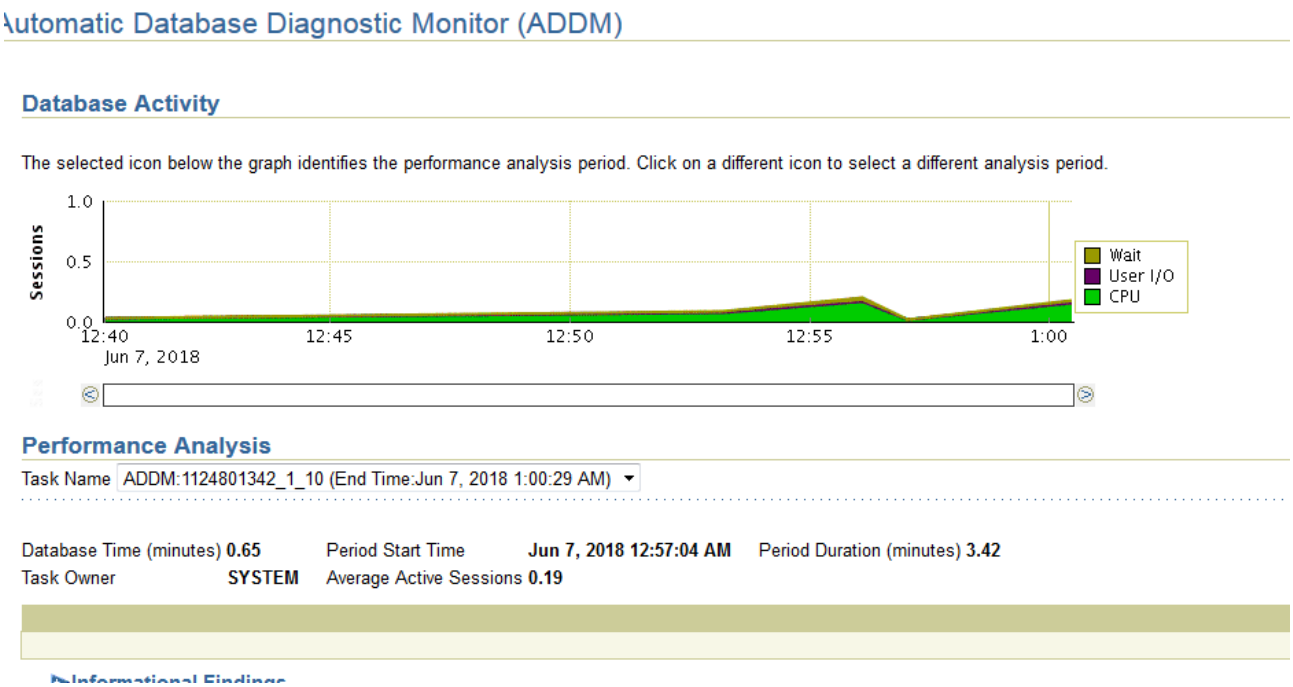
**4)**

-Repeat the command from step 2 to generate another snapshot.

```
SQL> EXEC dbms_workload_repository.create_snapshot;

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.82
```

**7)**

-select the radio button for the latest ADDM report, and click View Result.




**9)**

-Tidy up by dropping the TMPTAB table.

```
SQL> DROP TABLE tmptab;

Table dropped.

Elapsed: 00:00:00.36
```

*9.2.2)Configuring Alerts*

**4)**

-Scroll down to the “User Commits (per second)” alter, and set the warning and critical value to 1 and 4. These are artificially low thresholds that it will be simple to cross. Click OK to save this change.

| | | | | |
|---|---|---|---|---|
| User Calls (per second) | > | | | |
| User Calls (per transaction) | > | | | |
| User Commits (per second) | > | 1 | 4 | |
| User Commits (per transaction) | > | | | |

**5)**

-Connect to your database as user SYSTEM with SQL*Plus, and issue the COMMIT command a few times quickly.

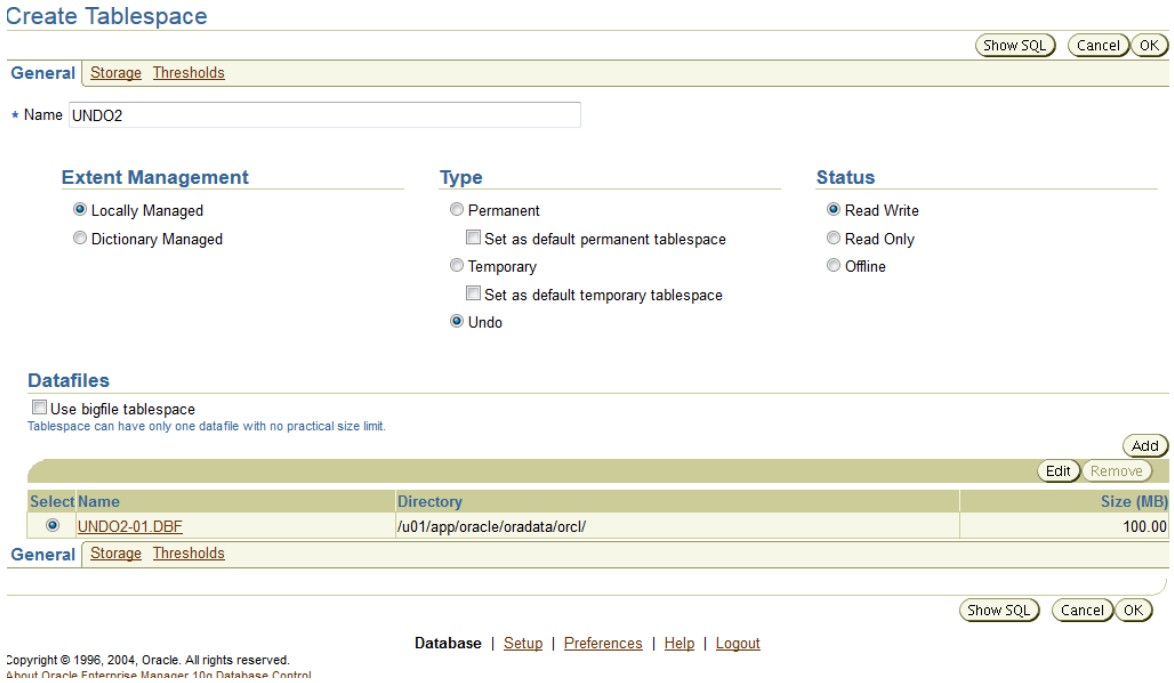
```
Commit complete.

Elapsed: 00:00:00.01
SQL> /

Commit complete.

Elapsed: 00:00:00.01
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.

Elapsed: 00:00:00.00
SQL> /

Commit complete.
```

## 10. Managing Undo

### 10.1)Creating an Undo Tablespace with Database Control

**4/5/6/7)**

-Enter UNDO2 as the tablespace name, and set the radio buttons to Extent Management “Locally Managed”, Type “Undo”, and Status “Read Write”.
-At the bottom of the screen, click Add to specify a datafile.
-Enter UNDO2-01.DBF as the File Name, leave everything else on default, and click Continue.
-On the Create Tablespace screen, click Show SQL, and study the statement used to create your undo tablespace. Click Return to return to the Create Tablespace screen, and click OK to create the tablespace.




**9)**

-Run this query, which will return one row for each tablespace in your database, and note that your new tablespace has contents UNDO, meaning that it can only be used for undo segments, and that retention is NOGUARANTEE, a topic covered shortly.

```
SQL> SELECT tablespace_name,contents,retention FROM dba_tablespaces;

TABLESPACE_NAME                CONTENTS  RETENTION
------------------------------ --------- -----------
SYSTEM                         PERMANENT NOT APPLY
UNDOTBS1                       UNDO      NOGUARANTEE
SYSAUX                         PERMANENT NOT APPLY
TEMP                           TEMPORARY NOT APPLY
USERS                          PERMANENT NOT APPLY
EXAMPLE                        PERMANENT NOT APPLY
INVENTORY                      PERMANENT NOT APPLY
UNDO2                          UNDO      NOGUARANTEE

8 rows selected.
```

**10)**

-Run this query, which will return one row for each rollback or undo segment in your database, and note that a number of undo segments have been created automatically in your new undo tablespace, but that they are all offline. Also note that the names of the automatic undo segments are in the form of “_SYSSMU*n*$”, where *n* is the undo segment number (usn).

```
SQL> SELECT tablespace_name,segment_name,status FROM dba_rollback_segs;

TABLESPACE_NAME                SEGMENT_NAME                   STATUS
------------------------------ ------------------------------ ---------------
SYSTEM                         SYSTEM                         ONLINE
UNDOTBS1                       _SYSSMU1$                      ONLINE
UNDOTBS1                       _SYSSMU2$                      ONLINE
UNDOTBS1                       _SYSSMU3$                      ONLINE
UNDOTBS1                       _SYSSMU4$                      ONLINE
UNDOTBS1                       _SYSSMU5$                      ONLINE
UNDOTBS1                       _SYSSMU6$                      ONLINE
UNDOTBS1                       _SYSSMU7$                      ONLINE
UNDOTBS1                       _SYSSMU8$                      ONLINE
UNDOTBS1                       _SYSSMU9$                      ONLINE
UNDOTBS1                       _SYSSMU10$                     ONLINE
UNDO2                          _SYSSMU11$                     OFFLINE
UNDO2                          _SYSSMU12$                     OFFLINE
UNDO2                          _SYSSMU13$                     OFFLINE
UNDO2                          _SYSSMU14$                     OFFLINE
UNDO2                          _SYSSMU15$                     OFFLINE
UNDO2                          _SYSSMU16$                     OFFLINE
UNDO2                          _SYSSMU17$                     OFFLINE
UNDO2                          _SYSSMU18$                     OFFLINE
UNDO2                          _SYSSMU19$                     OFFLINE
UNDO2                          _SYSSMU20$                     OFFLINE

21 rows selected.
```

## *10.2)Monitoring Undo With SQL*Plus*

**2)**

-Set up your session for displaying dates conveniently.

```
SQL> ALTER SESSION SET nls_date_format = 'dd/mm/yy hh24:mi:ss';

Session altered.

Elapsed: 00:00:00.05
SQL>
```

**3)**

-Query V$UNDOSTAT.

```
SQL> select begin_time,undoblks,maxquerylen,ssolderrcnt,nospaceerrcnt from v$undostat;

BEGIN_TIME          UNDOBLKS MAXQUERYLEN SSOLDERRCNT NOSPACEERRCNT
----------------- ---------- ----------- ----------- -------------
30/05/18 06:42:30         69           0           0             0
30/05/18 06:32:30         38           0           0             0
30/05/18 06:22:30         61           0           0             0
30/05/18 06:12:30         67           0           0             0
30/05/18 06:02:30         48           0           0             0
30/05/18 05:52:30        179           0           0             0
30/05/18 05:42:30        145           0           0             0
30/05/18 05:32:30         33           0           0             0
30/05/18 05:22:30        195           0           0             0
30/05/18 05:12:30      19574           0           0             0
30/05/18 05:02:30         49           0           0             0
30/05/18 04:52:30        127           0           0             0
30/05/18 04:42:30         61           0           0             0
30/05/18 04:32:30         43           0           0             0
30/05/18 04:22:30        115           0           0             0
30/05/18 04:12:30         72           0           0             0
30/05/18 04:02:30         27           0           0             0
30/05/18 03:52:30        157           0           0             0
30/05/18 03:42:30         61           0           0             0
30/05/18 03:32:30         36           0           0             0
30/05/18 03:22:30         69           0           0             0
30/05/18 03:12:30         59           0           0             0
30/05/18 03:02:30         44           0           0             0
30/05/18 02:52:30        146           0           0             0
30/05/18 02:42:30         71           0           0             0
30/05/18 02:32:30         32           0           0             0
30/05/18 02:22:30         73           0           0             0
30/05/18 02:12:30        408           0           0             0
30/05/18 02:02:30         91           0           0             0
30/05/18 01:52:30        305         150           0             0
```

**5)**

-Calculate the minimum necessary size in bytes for your undo tablespace that will prevent errors, given your current activity data.

```
SQL> SELECT(SELECT MAX(undoblks)/600 * MAX(maxquerylen) FROM v$undostat) * (SELECT value FROM v$parameter WHERE name='db_block_size') FROM dual;

(SELECTMAX(UNDOBLKS)/600*MAX(MAXQUERYLEN)FROMV$UNDOSTAT)*(SELECTVALUEFROMV$PARAM
--------------------------------------------------------------------------------
                                                                        40087552
```

## *11. Backup and Recovery*

### *11.1)Configuring the Database for Backup and Recovery*

**2)**

-Disable checkpoint tuning by setting the FAST_START_MTTR_TARGET parameter to zero.

```
SQL>
SQL> alter system set fast_start_mttr_target = 0;

System altered.

Elapsed: 00:00:00.04
SQL>
```

**3)**

-Simulate a workload by creating a table and starting a transaction.

```
SQL> CREATE TABLE t1 AS SELECT * FROM all_objects WHERE 1=2;

Table created.

Elapsed: 00:00:00.12
SQL> INSERT INTO t1 SELECT * FROM all_objects;

47750 rows created.

Elapsed: 00:00:02.35
SQL>
```

**4)**

-Run a query to see how much work would be required to recover the instance if it crashed right now.

```
SQL> SELECT recovery_estimated_ios,actual_redo_blks,estimated_mttr FROM v$instance_recovery;

RECOVERY_ESTIMATED_IOS ACTUAL_REDO_BLKS ESTIMATED_MTTR
---------------------- ---------------- --------------
                   649            17844             11

Elapsed: 00:00:00.04
```

**6)**

-Commit the transaction, and re-run the query from Step 3. Note that nothing much has changed: COMMIT has no effect on DBWn and will not advance the checkpoint position.

```
Elapsed: 00:00:00.12
SQL> commit;

Commit complete.

Elapsed: 00:00:00.02
SQL> SELECT recovery_estimated_ios,actual_redo_blks,estimated_mttr FROM v$instance_recovery;

RECOVERY_ESTIMATED_IOS ACTUAL_REDO_BLKS ESTIMATED_MTTR
---------------------- ---------------- --------------
                   699            18006             11

Elapsed: 00:00:00.06
SQL>
```

**7)**

-Issue a manual checkpoint.

```
SQL> ALTER SYSTEM checkpoint;

System altered.

Elapsed: 00:00:00.11
```

**8)**

-Re-run the query from Step 4. Note that the RECOVERY_ESTIMATED_IOS and ACTUAL _REDO_BLKS columns have dropped substantially, perhaps to zero. The ESTIMATED_MTTR column may not have reduced, because this column is not updated in real time.

```
SQL> SELECT recovery_estimated_ios,actual_redo_blks,estimated_mttr FROM v$instance_recovery;

RECOVERY_ESTIMATED_IOS ACTUAL_REDO_BLKS ESTIMATED_MTTR
---------------------- ---------------- -------------
                    20               23            10

Elapsed: 00:00:00.07
```

**9)**

-Tidy up by dropping the table.

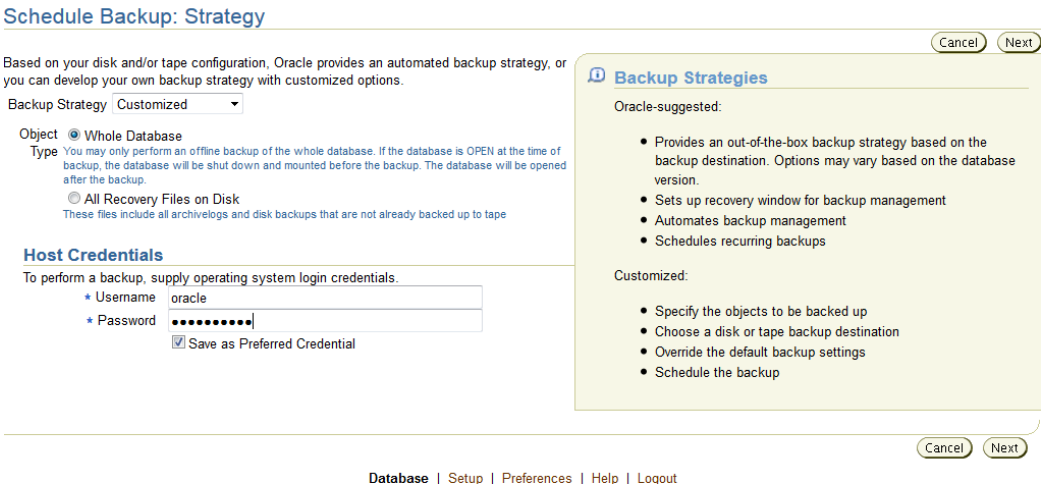
```
SQL> DROP TABLE t1;

Table dropped.

Elapsed: 00:00:00.36
```

## *11.2) Backing Up an Oracle Database*

### *11.2.1)Part One*

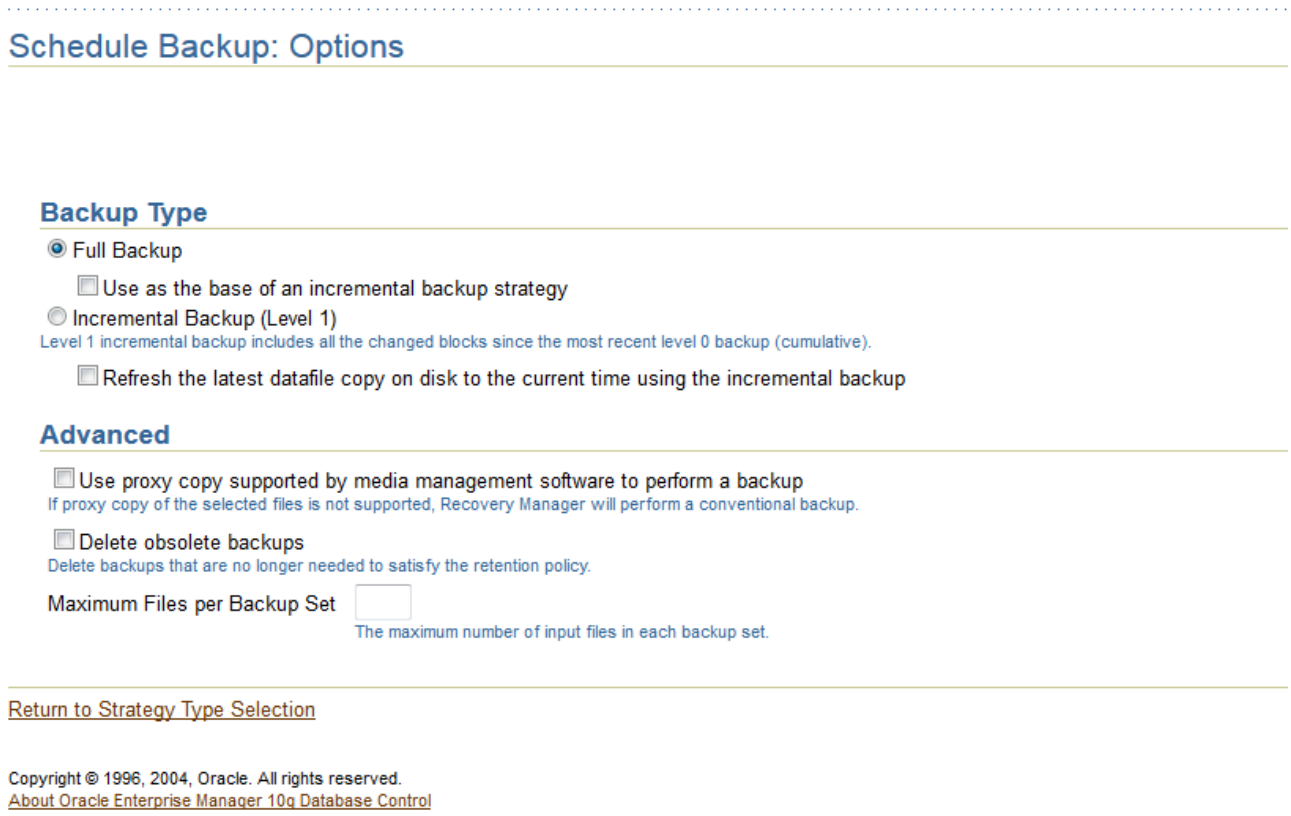
**2)**

-In the Backup Strategy drop-down box, select Customized and click the Whole Database radio button. In the Host Credentials section, enter an operating system username and password. Click Next to reach the Schedule Backup: Options window.



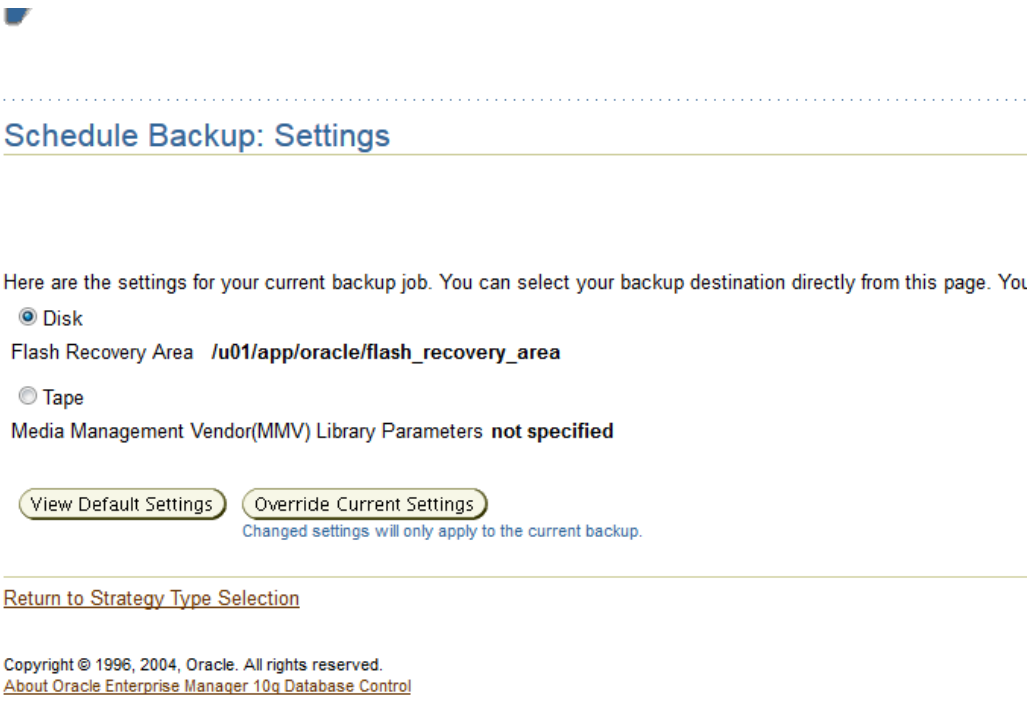
**3)**

-Leave everything on defaults: a full, online backup with all archive logs. Click Next to reach the Schedule Backup: Settings window.



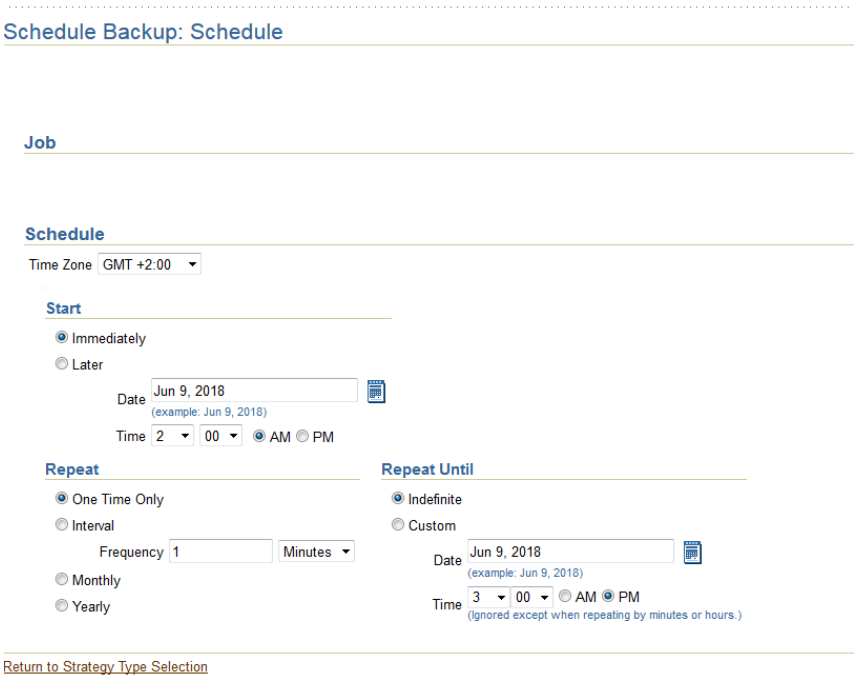

**4)**

-Leave everything on default to schedule a disk backup to your flash recovery area directory. Click Next to reach the Schedule Backup: Schedule window.



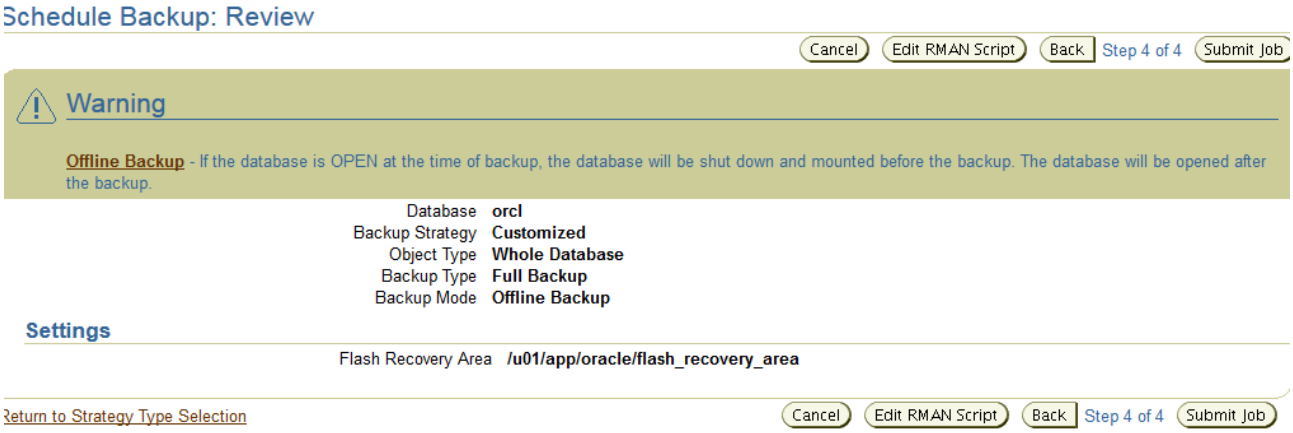
**5)**

-Leave everything on default to run the backup immediately as a one-off job. Click Next to reach the Schedule Backup: Review window.



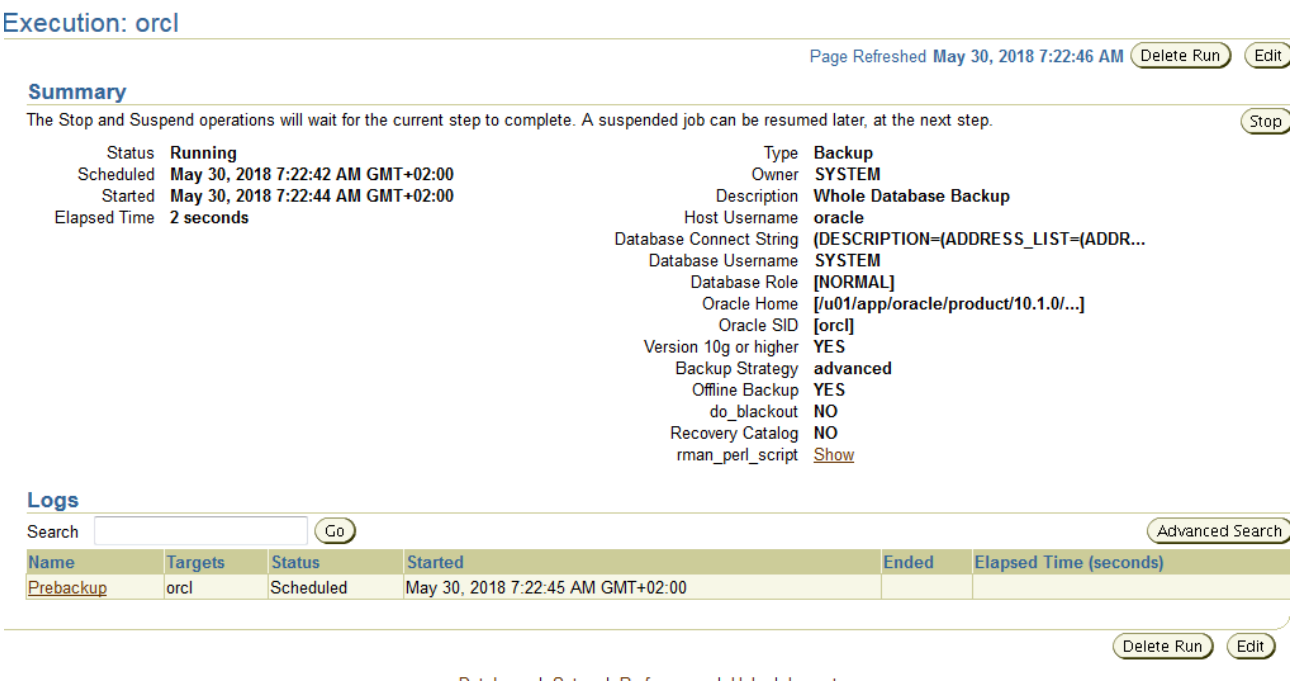

**6)**

-Click the Submit Job button to launch the backup.

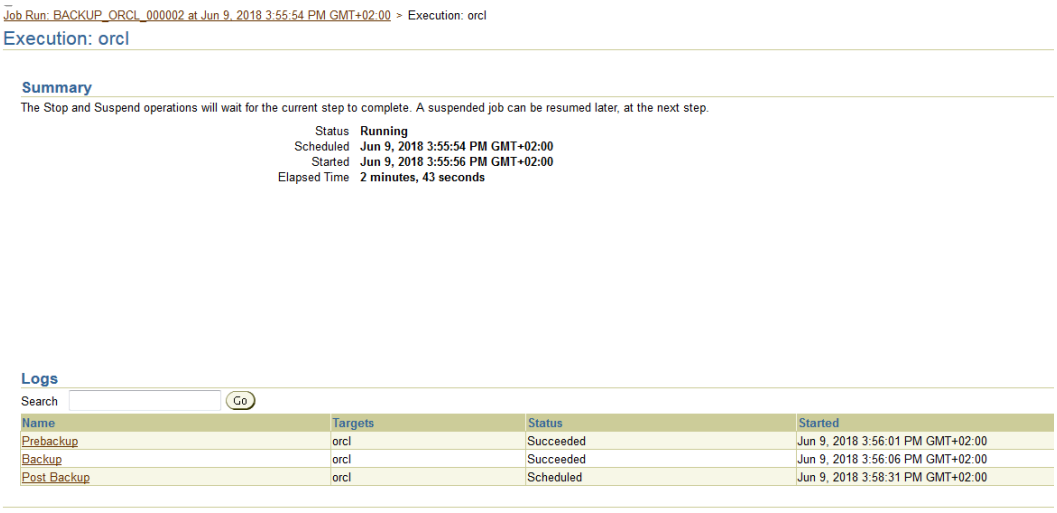


7)

-Click the View Job button to check how the job is running, and then refresh the browser window to monitor progress.







*11.2.2)Part Two*

**2)**

-Issue this command:

```
With the Partitioning, OLAP and Data Mining options

SQL> ALTER DATABASE BACKUP controlfile TO trace;

Database altered.
```

**3)**

-Locate your user dump destination.

```
SQL> show parameters user_dump_dest;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
user_dump_dest                       string      /u01/app/oracle/admin/orcl/udu
                                                 mp
```

**5)**

-Identify the newest file in the directory.Open the trace file with any editor you please and study the contents.

```
-rw-r-----    1 oracle   oinstall      785 May 30 07:23 orcl_ora_833.trc
-rw-r-----    1 oracle   oinstall      638 May 30 07:23 orcl_ora_1069.trc
-rw-r-----    1 oracle   oinstall     1730 May 30 07:26 orcl_ora_1088.trc
-rw-r-----    1 oracle   oinstall     6720 May 30 21:07 orcl_ora_21032.trc
```

*11.3)Recovering Oracle Databases*

*11.3.1)Part One*

**1)**

-Connect to your database with SQL*Plus, and ensure that your controlfile is multiplexed.

```
SQL> SELECT * FROM v$controlfile;

STATUS
-------
NAME
--------------------------------------------------------------------------------
IS_
---

/u01/app/oracle/oradata/orcl/control01.ctl
NO


/u01/app/oracle/oradata/orcl/control02.ctl
NO


/u01/app/oracle/oradata/orcl/control03.ctl
NO
```

**3)**

-Issue a startup command. The startup will stop in nomount mode, with an “ORA-00205: error in identifying controlfile, check alert log for more info” error message.

```
Enter user-name: sys as sysdba
Enter password:
Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  167772160 bytes
Fixed Size                  778212 bytes
Variable Size             66068508 bytes
Database Buffers         100663296 bytes
Redo Buffers                262144 bytes
ORA-00205: error in identifying controlfile, check alert log for more info
```

**4)**

-Copy your surviving controlfile to the name and location of the file you renamed.

```
[oracle@vmware orcl]$ ls
control01.ctl.old  inventory01.dbf  lab_12_a.log  lab_12_f.log  sysaux01.dbf  undotbs01.dbf
control02.ctl      lab_12_a.bad     lab_12_f.bad  redo01.log    system01.dbf  users01.dbf
control03.ctl      lab_12_a.ctl     lab_12_f.ctl  redo02.log    temp01.dbf
example01.dbf      lab_12_a.dat     lab_12_f.dat  redo03.log    UNDO2-01.DBF
[oracle@vmware orcl]$ mv control01.ctl.old control01.ctl
[oracle@vmware orcl]$ sqlplus
```

**5)**

-Issue another startup command, which will be successful.

```
Enter user-name: sys as sysdba
Enter password:
Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  167772160 bytes
Fixed Size                  778212 bytes
Variable Size             66068508 bytes
Database Buffers         100663296 bytes
Redo Buffers                262144 bytes
Database mounted.
Database opened.
```

*11.3.2)Part Two*

**2)**

-Observe the state of your online logs.

```
Elapsed: 00:00:00.01
SQL> SELECT group#,status,member FROM v$logfile ORDER BY group

    GROUP# STATUS
---------- -------
MEMBER
------------------------------------------------------------
         1
/u01/app/oracle/oradata/orcl/redo01.log

         1
/u01/app/oracle/oradata/orcl/redo01B.log

         2
/u01/app/oracle/oradata/orcl/redo02.log

         2
/u01/app/oracle/oradata/orcl/redo02B.log

         3
/u01/app/oracle/oradata/orcl/redo03.log

         3
/u01/app/oracle/oradata/orcl/redo03B.log


6 rows selected.

Elapsed: 00:00:00.00
```

**4)**

-Shut down the database.

```
Elapsed: 00:00:00.00
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

**5)**

-Using an operating system command, simulate media failure by deleting one of the members.

```
[oracle@vmware orcl]$ ls
control01.ctl  control03.ctl  redo01.bkp   redo02B.log  redo03B.log  sysaux01.dbf  temp01.dbf     users01.dbf
control02.ctl  example01.dbf  redo01B.log  redo02.log   redo03.log   system01.dbf  undotbs01.dbf
[oracle@vmware orcl]$
```

**6)**

-Start up the database and simulate user activity by performing a few log switches.

```
SQL> ALTER SYSTEM switch logfile;

System altered.

Elapsed: 00:00:05.13
SQL> /

System altered.

Elapsed: 00:00:00.02
SQL> /

System altered.

Elapsed: 00:00:02.81
```

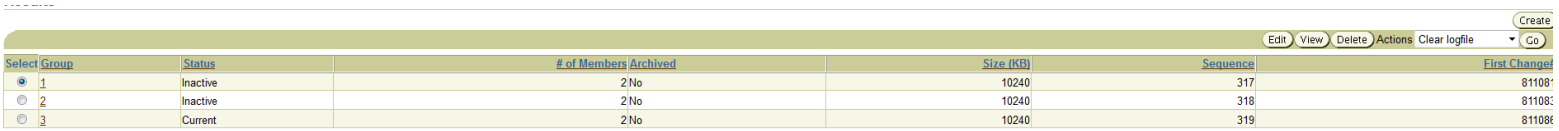
**7)**

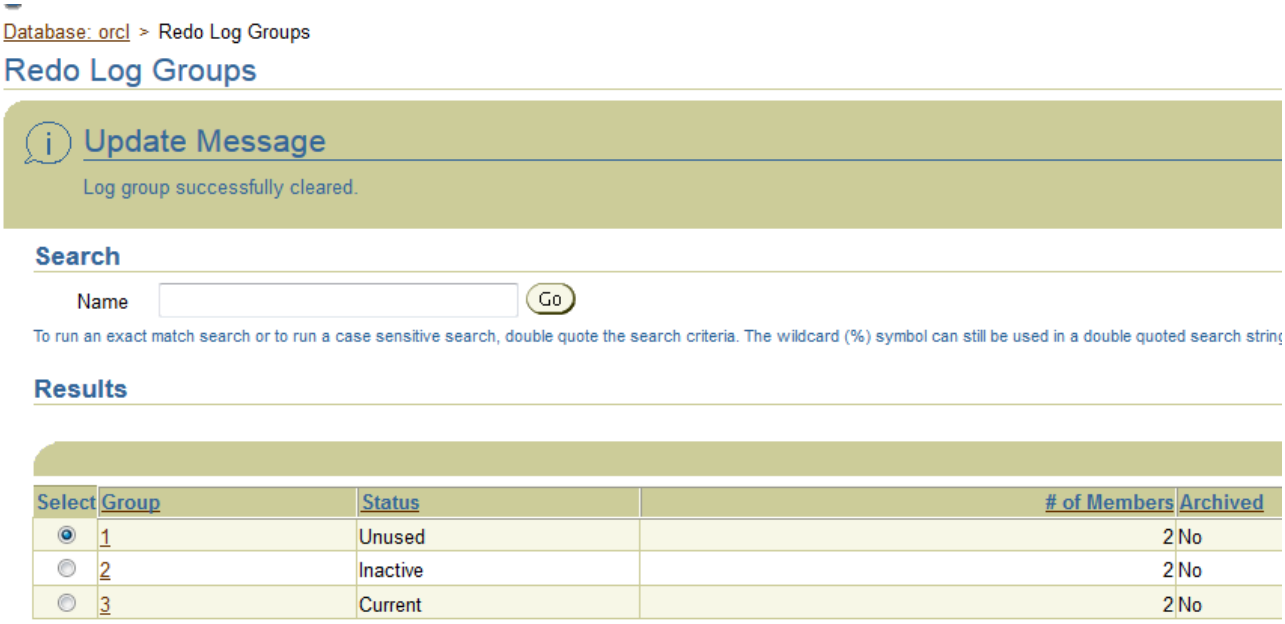
-Check the state of your logfile members.

```
SQL> SELECT group#,status,member FROM v$logfile ORDER BY group#

    GROUP# STATUS
---------- -------
MEMBER
------------------------------------------------------------
         1 INVALID
/u01/app/oracle/oradata/orcl/redo01.log

         1
/u01/app/oracle/oradata/orcl/redo01B.log

         2
/u01/app/oracle/oradata/orcl/redo02.log

         2
/u01/app/oracle/oradata/orcl/redo02B.log

         3
/u01/app/oracle/oradata/orcl/redo03.log

         3
/u01/app/oracle/oradata/orcl/redo03B.log


6 rows selected.
```

**11)**

-Clear the logfile group by selecting its radio button using the Clear Logfile choice in the Actions drop-down list, and clicking Go.




**12)**

-In your SQL*Plus session, confirm that the problem has been fixed.

```
Elapsed: 00:00:00.01
SQL> SELECT group#,status,member FROM v$logfile ORDER BY group#;

    GROUP# STATUS
---------- -------
MEMBER
--------------------------------------------------------------------------------
         1
/u01/app/oracle/oradata/orcl/redo01.log

         1
/u01/app/oracle/oradata/orcl/redo01B.log

         2
/u01/app/oracle/oradata/orcl/redo02.log

         2
/u01/app/oracle/oradata/orcl/redo02B.log

         3
/u01/app/oracle/oradata/orcl/redo03.log

         3
/u01/app/oracle/oradata/orcl/redo03B.log


6 rows selected.

Elapsed: 00:00:00.02
```

*11.3.3)Part Three*

**1)**

-Connect to your database as user SYSTEM using SQL*Plus, and create a tablespace.

```
Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> create tablespace noncrit datafile
  2  '/u01/app/oracle/oradata/orcl/noncrit.dbf' size 2m;

Tablespace created.
```

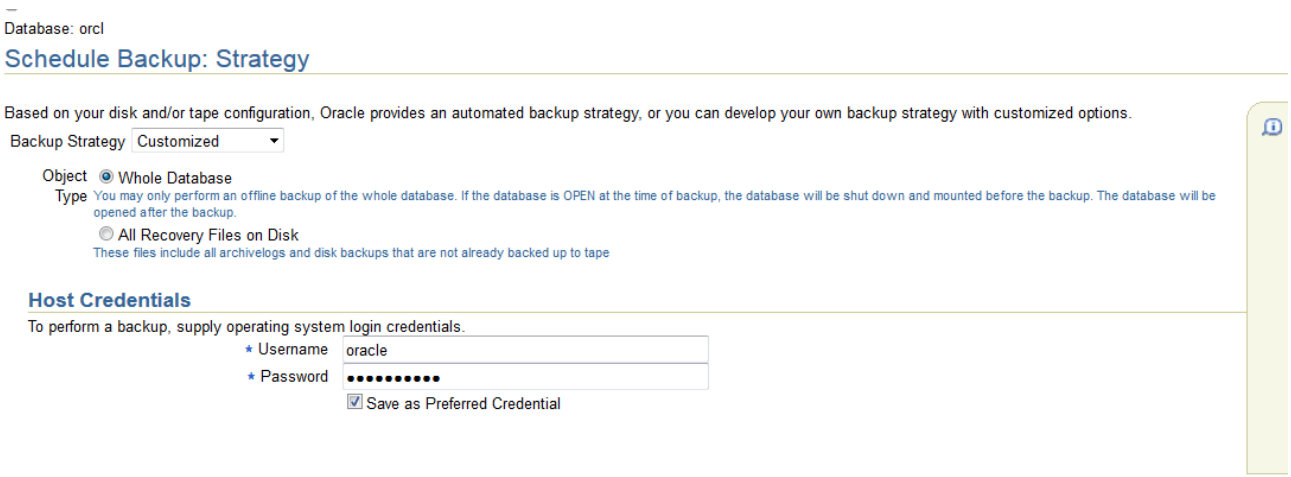
**2)**

-Create a table within the new tablespace and insert a row into it.

```
Elapsed: 00:00:00.20
SQL> create table ex1133 (c1 date) tablespace noncrit;

Table created.

Elapsed: 00:00:00.10
SQL> insert into ex1133 values(sysdate);

1 row created.

Elapsed: 00:00:00.00
SQL> commit;

Commit complete.
```

**5)**

-In the Schedule Backup: Strategy window, select Customized in the Backup Strategy drop-down box.




A partir de ce point, je n'ai pas pu malheureusement trouver une manière de faire un backup des tablespaces. Comme on peut le voir sur la capture d'écran, je n'ai pas d'option pour effectuer la sauvegarde des tablespaces.

Je me suis donc permis de faire un peu de recherche afin de trouver une manière de corriger ce problème. J'ai cru comprendre qu'il était possible de configurer ses propres backups avec le lien "Configure Backup Settings":


Mais je n'ai pas réussi à trouver une manière de configurer un backup des tablespaces via cette option là.

Cependant, comme on peut voir dans la partie 12 "Recovery manager", il est possible de faire un backup du tablespace avec RMAN et la commande "backup tablespace NONCRIT".

*12. recovery manager*

*12.1)Recovery Manager Configuration*

**1)**

-Connect to your database as the target database in the default NOCATALOG mode as the SYSTEM user.

```
[oracle@vmware oradata]$ rman TARGET system/oracle NOCATALOG

Recovery Manager: Release 10.1.0.3.0 - Production

Copyright (c) 1995, 2004, Oracle.  All rights reserved.

connected to target database: ORCL (DBID=1124801342)
using target database controlfile instead of recovery catalog
```

**2)**

-Use the **RMAN SHOW ALL** command to generate a listing of the **RMAN** configuration settings.

```
RMAN> SHOW ALL;

RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO '/u01/app/oracle/product/10.1.0/db_1/dbs/snapcf_orcl.f';
# default
```

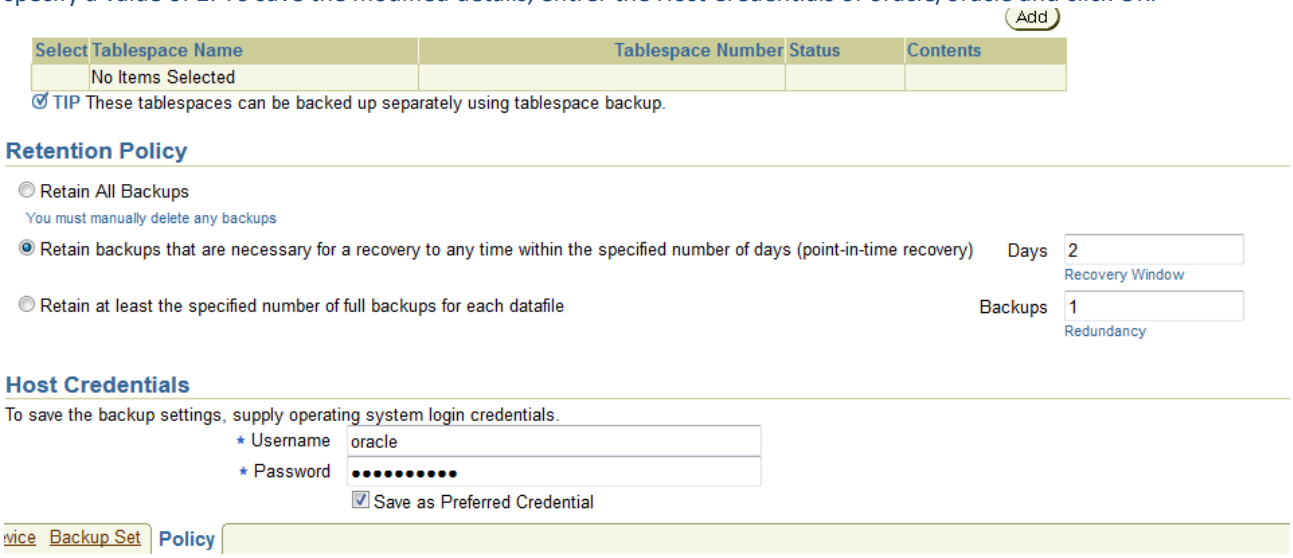
**3)**

-Configure **RMAN** to automatically back up the control file and SPFILE whenever a backup of the database or data files is taken.

```
# default

RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored
```

**9)**

-Choose "Retain backups that are necessary for a recovery to any time within the specified number of days and specify a value of 2. To save the modified details, entrer the Host Credentials of oracle/oracle and click OK.




**10)**

-Verify the backup retention policy setting using the **RMAN** utility and the **SHOW** command.

```
RMAN> SHOW RETENTION POLICY;

RMAN configuration parameters are:
CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 2 DAYS;

RMAN>
```

## *12.2)Using Recovery Manager*

**4)**

-To check using SQL*Plus, use the ARCHIVE LOG LIST command.

```
SQL> ARCHIVE LOG LIST
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     30
Current log sequence           32
```

**5)**

-The database is not currently archiving. Correct this problem with the following commands, or use Enterprise Manager.

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

Total System Global Area  167772160 bytes
Fixed Size                   778212 bytes
Variable Size              66068508 bytes
Database Buffers          100663296 bytes
Redo Buffers                 262144 bytes
Database mounted.
SQL> alter database archivelog:
  2
SQL> alter database archivelog;

Database altered.

Elapsed: 00:00:00.03
SQL> alter database open;

Database altered.

Elapsed: 00:00:00.49
SQL> archive log list;
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     30
Next log sequence to archive   32
Current log sequence           32
SQL>
```

**7)**

-Use the RMAN REPORT command to generate a listing of your database structure.

```
RMAN> REPORT SCHEMA;

Report of database schema
File K-bytes    Tablespace           RB segs Datafile Name
---- ---------- -------------------- ------- ------------------
1        450560 SYSTEM               ***     /u01/app/oracle/oradata/orcl/system01.dbf
2         30720 UNDOTBS1             ***     /u01/app/oracle/oradata/orcl/undotbs01.dbf
3        225280 SYSAUX               ***     /u01/app/oracle/oradata/orcl/sysaux01.dbf
4          5120 USERS                ***     /u01/app/oracle/oradata/orcl/users01.dbf
5        153600 EXAMPLE              ***     /u01/app/oracle/oradata/orcl/example01.dbf
6          2048 NONCRIT              ***     /u01/app/oracle/product/10.1.0/oradata/orcl/noncrit
.dbf

RMAN>
```

**8)**

-Obtain a listing of all database backup sets that currently exist.

```
RMAN> LIST BACKUP OF DATABASE;


RMAN> LIST COPY OF DATABASE;


RMAN>
```

**9)**

-Use RMAN to back up the data files belonging to the EXAMPLE and USERS tablespaces. Be sure you also make a copy of the current control file and server parameter file.
Your backups should be placed in the $HOME/DONTTOUCH/ directory and should use the format df_%d_%s_ %p.bak for the file names.

```
RMAN> BACKUP AS BACKUPSET
2> FORMAT '$HOME/DONTTOUCH/df_%d_%s_%p.bak'
3> TABLESPACE USERS, EXAMPLE;

Starting backup at 31-MAY-18
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00005 name=/u01/app/oracle/oradata/orcl/example01.dbf
input datafile fno=00004 name=/u01/app/oracle/oradata/orcl/users01.dbf
channel ORA_DISK_1: starting piece 1 at 31-MAY-18
channel ORA_DISK_1: finished piece 1 at 31-MAY-18
piece handle=/home/oracle/DONTTOUCH/df_ORCL_4_1.bak comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 31-MAY-18

Starting Control File and SPFILE Autobackup at 31-MAY-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/autobackup/2018_05_31/o1_mf_s_977603265_f
0ho1c0_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 31-MAY-18
```

**10)**

-Create an image copy of two data files. Use the following information:

- Copy the SYSTEM tablespace and name the copy sys01.cpy with a tag of SYSTEM01
- Copy the SYSAUX tablespace and name the copy sysaux01.cpy with a tag of SYSAUX01
- The files should be written to the Flash Recovery Area.

```
RMAN> BACKUP AS COPY
2> FORMAT 'sys01.cpy'
3> TABLESPACE SYSTEM
4> TAG=SYSTEM01;

Starting backup at 31-MAY-18
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile fno=00001 name=/u01/app/oracle/oradata/orcl/system01.dbf
output filename=/u01/app/oracle/product/10.1.0/db_1/dbs/sys01.cpy tag=SYSTEM01 recid=2 stamp=977
603354
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
Finished backup at 31-MAY-18

Starting Control File and SPFILE Autobackup at 31-MAY-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/autobackup/2018_05_31/o1_mf_s_977603354_fk
0hqty1_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 31-MAY-18

RMAN> BACKUP AS COPY
2> FORMAT 'sysaux01.cpy'
3> TABLESPACE SYSAUX
4> TAG=SYSAUX01;

Starting backup at 31-MAY-18
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile fno=00003 name=/u01/app/oracle/oradata/orcl/sysaux01.dbf
output filename=/u01/app/oracle/product/10.1.0/db_1/dbs/sysaux01.cpy tag=SYSAUX01 recid=3 stamp=
977603416
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
Finished backup at 31-MAY-18

Starting Control File and SPFILE Autobackup at 31-MAY-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/autobackup/2018_05_31/o1_mf_s_977603417_fk
0hst09_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 31-MAY-18

RMAN>
```

**11)**

-Obtain a listing of all database files that have not been backed up.

```
RMAN> REPORT NEED BACKUP;

RMAN retention policy will be applied to the command
RMAN retention policy is set to recovery window of 2 days
Report of files whose recovery needs more than 2 days of archived logs
File Days  Name
---- ----- -----------------------------------------------------
2    5042  /u01/app/oracle/oradata/orcl/undotbs01.dbf
```

**12)**

-Take a full backup of the database, including archived logs. Use as little space as possible to store the backup.

```
RMAN> BACKUP AS COMPRESSED BACKUPSET DATABASE PLUS ARCHIVELOG;


Starting backup at 31-MAY-18
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed archive log backupset
channel ORA_DISK_1: specifying archive log(s) in backup set
input archive log thread=1 sequence=32 recid=1 stamp=977603525
channel ORA_DISK_1: starting piece 1 at 31-MAY-18
channel ORA_DISK_1: finished piece 1 at 31-MAY-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/backupset/2018_05_31/o1_mf_annnn_TAG201805
31T203205_fk0hx6x3_.bkp comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 31-MAY-18

Starting backup at 31-MAY-18
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001 name=/u01/app/oracle/oradata/orcl/system01.dbf
input datafile fno=00003 name=/u01/app/oracle/oradata/orcl/sysaux01.dbf
input datafile fno=00005 name=/u01/app/oracle/oradata/orcl/example01.dbf
input datafile fno=00002 name=/u01/app/oracle/oradata/orcl/undotbs01.dbf
input datafile fno=00004 name=/u01/app/oracle/oradata/orcl/users01.dbf
input datafile fno=00006 name=/u01/app/oracle/product/10.1.0/oradata/orcl/noncrit.dbf
channel ORA_DISK_1: starting piece 1 at 31-MAY-18
channel ORA_DISK_1: finished piece 1 at 31-MAY-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/backupset/2018_05_31/o1_mf_nnndf_TAG201805
31T203207_fk0hx81p_.bkp comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:25
Finished backup at 31-MAY-18

Starting backup at 31-MAY-18
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting compressed archive log backupset
channel ORA_DISK_1: specifying archive log(s) in backup set
input archive log thread=1 sequence=33 recid=2 stamp=977603553
channel ORA_DISK_1: starting piece 1 at 31-MAY-18
channel ORA_DISK_1: finished piece 1 at 31-MAY-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/backupset/2018_05_31/o1_mf_annnn_TAG201805
31T203233_fk0hy2ff_.bkp comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:02
Finished backup at 31-MAY-18

Starting Control File and SPFILE Autobackup at 31-MAY-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/autobackup/2018_05_31/o1_mf_s_977603555_fk
0hy3gb_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 31-MAY-18
```

## 13. Managing Data Recovery

### 13.1)Recovering from Noncritical Losses

**1)**

-Get the name of the default temporary tablespace from the DATABASE_PROPERTIES view and the data files associated with this tablespace from DBA_TEMP_FILES.

```
SQL> SELECT * FROM database_properties
  2  WHERE property_name like '%TEMP%';

PROPERTY_NAME
------------------------------
PROPERTY_VALUE
--------------------------------------------------------------------------------
DESCRIPTION
--------------------------------------------------------------------------------
DEFAULT_TEMP_TABLESPACE
TEMP
Name of default temporary tablespace


Elapsed: 00:00:00.02
SQL> SELECT file_name FROM dba_temp_files WHERE tablespace_name = 'TEMP';

FILE_NAME
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/orcl/temp01.dbf

Elapsed: 00:00:00.02
SQL>
```

**2)**

-Delete the temporary tablespace data files at the operating system level.

```
[oracle@vmware oradata]$ rm /u01/app/oracle/oradata/orcl/temp01.dbf
[oracle@vmware oradata]$
```

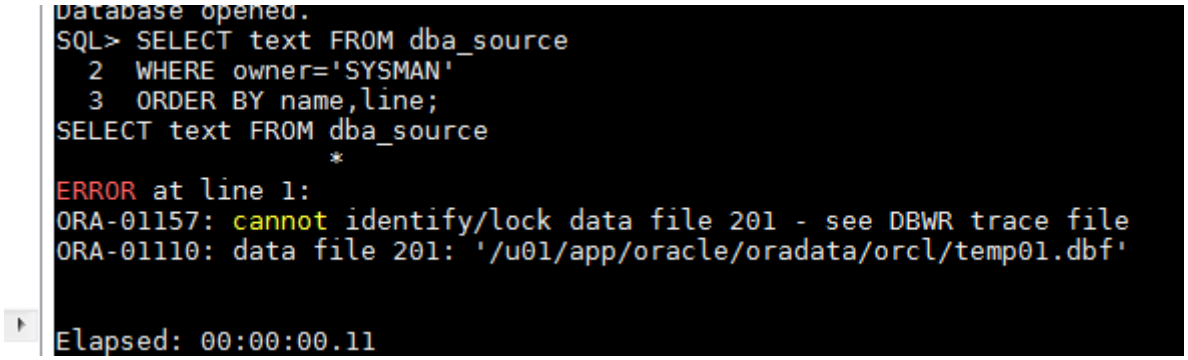
**3)**

-Connect to the database as a SYSDBA user, shutdown the instance, and restart it.

```
Copyright (c) 1982, 2004, Oracle.  All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 10g Enterprise Edition Release 10.1.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> shutdozn immediate;
SP2-0734: unknown command beginning "shutdozn i..." - rest of line ignored.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup;
ORACLE instance started.

Total System Global Area  167772160 bytes
Fixed Size                   778212 bytes
Variable Size              66068508 bytes
Database Buffers          100663296 bytes
Redo Buffers                 262144 bytes
Database mounted.
Database opened.
SQL>
```

**4)**

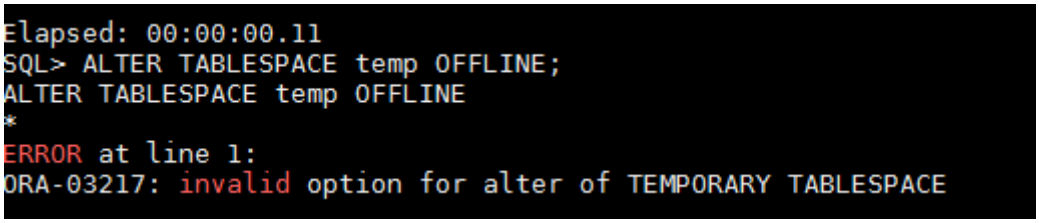
-Perform a query against a table in the database that involves sorting of data. What happens?




La table dba_source, comme on a pu voir dans la question 1, à comme pour fichier temporaire /u01/app/oracle/oradata/orl/temp01.dbf. Or, le fichier à été supprimé. L'exécution de la requête à donc été interrompu à cause de l'abscence de ce fichier.

**5)**

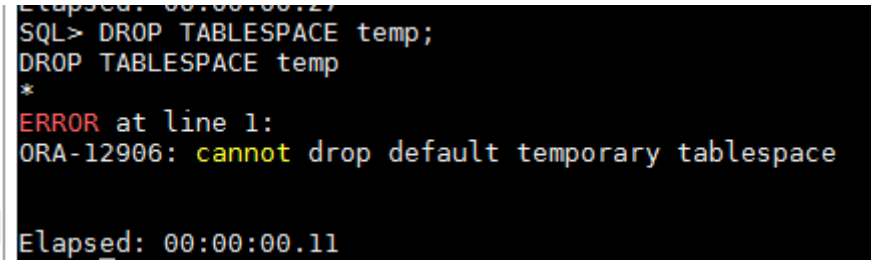
Attempt to take the temporary tablespace offline before recovering it. What happens?




Il semblerait qu'il n'arrive pas à reconnaître la tablespace "temp". Cela doit être le tablespace contenu dans le fichier temp01.dbf.

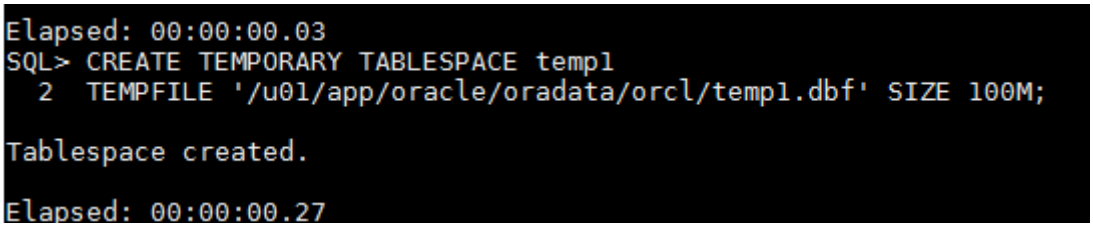
**6)**

-Drop the temporary tablespace. What happens?




**7)**

-Create a new temporary tablespace named TEMP1 containing a single data file named temp1.dbf which is 100 MB in size.



**8)**

-Change the database default temporary tablespace to TEMP1.

```
Elapsed: 00:00:00.11
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp1;

Database altered.
```

**9)**

-Retry your query that involved a sort operation. What happens now?

```
SQL> SELECT text FROM dba_source
  2  WHERE owner='SYSMAN'
  3  ORDER BY name,type,line;

TEXT
--------------------------------------------------------------------------------
TRIGGER blackout_change
    BEFORE INSERT ON MGMT_BLACKOUT_STATE FOR EACH ROW
DECLARE

    latest_availability_rowid ROWID;
    latest_availability_status NUMBER;
    latest_availability_sev_guid RAW(16);
    new_annotation_guid RAW(16);
    l_emd_url VARCHAR2(2000);
    l_count NUMBER;
    l_target_guid RAW(16);
    is_target_type_host BOOLEAN;
    host_guid RAW(16);
    host_availability_status NUMBER;
    l_current_target_state NUMBER;

    l_current_blackout_state NUMBER;

    l_created_thru MGMT_BLACKOUTS.created_thru%TYPE;

    l_actual_start_time DATE;
    l_actual_end_time DATE;
    l_occurrence_number NUMBER;

    response_metric_guid RAW(16);
    update_availability BOOLEAN := false;
```

**10)**

-Drop the temporary tablespace with the missing data files. You must remove the tablespace and the file associated using a single SQL command.

```
Elapsed: 00:00:02.02
SQL> DROP TABLESPACE temp INCLUDING CONTENTS AND DATAFILES;

Tablespace dropped.

Elapsed: 00:00:00.24
```

**11)**

-Perform a backup of the database.

```
RMAN> backup database;

Starting backup at 02-JUN-18
using target database controlfile instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=133 devtype=DISK
channel ORA_DISK_1: starting full datafile backupset
channel ORA_DISK_1: specifying datafile(s) in backupset
input datafile fno=00001 name=/u01/app/oracle/oradata/orcl/system01.dbf
input datafile fno=00003 name=/u01/app/oracle/oradata/orcl/sysaux01.dbf
input datafile fno=00005 name=/u01/app/oracle/oradata/orcl/example01.dbf
input datafile fno=00002 name=/u01/app/oracle/oradata/orcl/undotbs01.dbf
input datafile fno=00004 name=/u01/app/oracle/oradata/orcl/users01.dbf
input datafile fno=00006 name=/u01/app/oracle/product/10.1.0/oradata/orcl/noncrit.dbf
channel ORA_DISK_1: starting piece 1 at 02-JUN-18
channel ORA_DISK_1: finished piece 1 at 02-JUN-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/backupset/2018_06_02/o1_mf_nnndf_TAG201806
02T225759_fk616qbx_.bkp comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 02-JUN-18

Starting Control File and SPFILE Autobackup at 02-JUN-18
piece handle=/u01/app/oracle/flash_recovery_area/ORCL/autobackup/2018_06_02/o1_mf_s_977785094_fk
6176fy_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 02-JUN-18
```

## *13.2) Database Recovery*

### *13.2.1)Part One*

**1)**

-As user system/oracle create the table HR.DEPARTMENTS2 by selecting all rows from the HR.DEPARTMENTS table. Confirm that the new table exists, and record the total number of rows in the table.
View the active log by querying V$LOG. Perform a log switch when finished.

```
SQL> connect system/oracle as sysdba
Connected.
SQL> CREATE TABLE HR.DEPARTMENTS2 AS SELECT * FROM HR.DEPARTMENTS;

Table created.

Elapsed: 00:00:00.08
SQL> select count(*) from hr.departments2;

  COUNT(*)
----------
        27

Elapsed: 00:00:00.00
SQL> select sequence#,status from v$log;

 SEQUENCE# STATUS
---------- ----------------
        77 INACTIVE
        78 INACTIVE
        79 CURRENT

Elapsed: 00:00:00.02
SQL> alter system switch logfile;

System altered.

Elapsed: 00:00:00.27
```

**2)**

-Check and record the system time and date.

```
Elapsed: 00:00:00.27
SQL> !date
Sat Jun  2 23:10:01 CEST 2018
```

**3)**

-Query V$LOG again to confirm the switch and then insert three lines into the HR.DEPARTMENTS2 table and commit. Confirm the number of row in the table. These INSERTs represent the introduction of questionable data into the table.

```
QL> select sequence#, status from v$log;

SEQUENCE# STATUS
--------- ----------------
       80 CURRENT
       78 INACTIVE
       79 ACTIVE

lapsed: 00:00:00.00
QL> select count(*) from hr.departments2;

 COUNT(*)
---------
       27

lapsed: 00:00:00.00
```

**4)**

-Shutdown the database, and restart it in mount mode.

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

Total System Global Area  167772160 bytes
Fixed Size                   778212 bytes
Variable Size              66068508 bytes
Database Buffers          100663296 bytes
Redo Buffers                 262144 bytes
Database mounted.
```

**5)**

-Using RMAN, recover the database to a point in time before the new data was introduced using the information you recorded before the inserts were performed.

```
RMAN> run {
2> set until time "TO_DATE('18-JUN-02:23:10:01','YY-MON-DD:HH24:MI:SS')";
3> restore database;
4> recover database;
5> }

executing command: SET until clause

Starting restore at 02-JUN-18
allocated channel: ORA_DISK_1
channel ORA_DISK_1: sid=160 devtype=DISK

channel ORA_DISK_1: starting datafile backupset restore
channel ORA_DISK_1: specifying datafile(s) to restore from backup set
restoring datafile 00001 to /u01/app/oracle/oradata/orcl/system01.dbf
restoring datafile 00002 to /u01/app/oracle/oradata/orcl/undotbs01.dbf
restoring datafile 00003 to /u01/app/oracle/oradata/orcl/sysaux01.dbf
restoring datafile 00004 to /u01/app/oracle/oradata/orcl/users01.dbf
restoring datafile 00005 to /u01/app/oracle/oradata/orcl/example01.dbf
restoring datafile 00006 to /u01/app/oracle/product/10.1.0/oradata/orcl/noncrit.dbf
```

**6)**

-Open the database with the RESETLOGS option and confirm the recovery.

```
SQL> alter database open resetlogs;

Database altered.

Elapsed: 00:00:01.14
SQL> select count(*) from hr.departments2;

  COUNT(*)
----------
        27

Elapsed: 00:00:00.01
```

*13.2.1)Part Two*

**1)**

-Determine the current log sequence and write it down.

```
Elapsed: 00:00:00.01
SQL> select sequence#,status from v$log;

 SEQUENCE# STATUS
---------- ----------------
         1 CURRENT
         0 UNUSED
         0 UNUSED

Elapsed: 00:00:00.02
```

**2)**

-Verify the row count for the HR.DEPARTMENTS2 table.

```
SQL> select count(*) from hr.departments2;

  COUNT(*)
----------
        27

Elapsed: 00:00:00.02
```

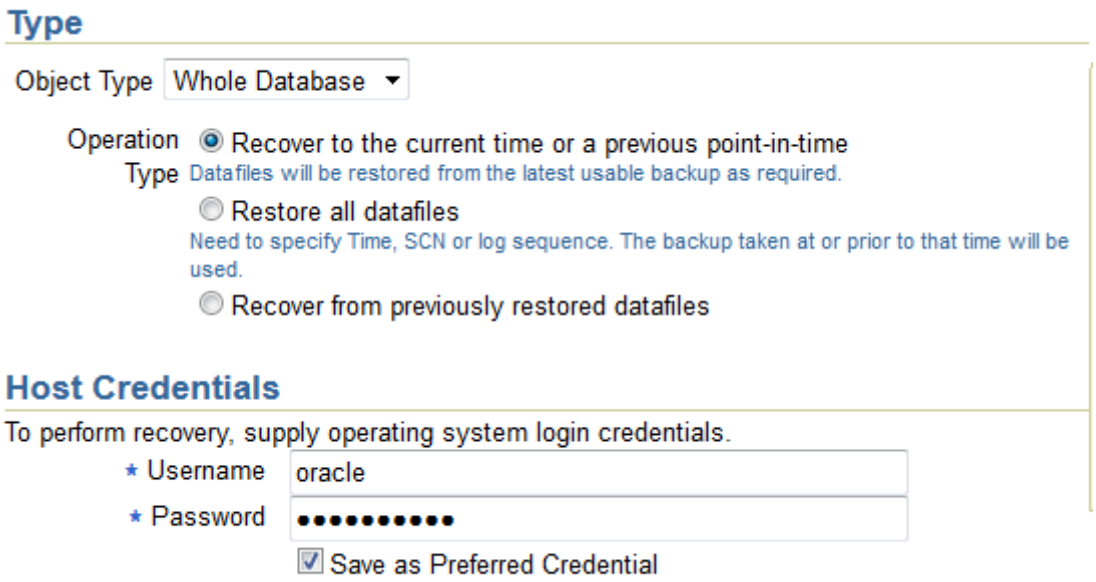
**3)**

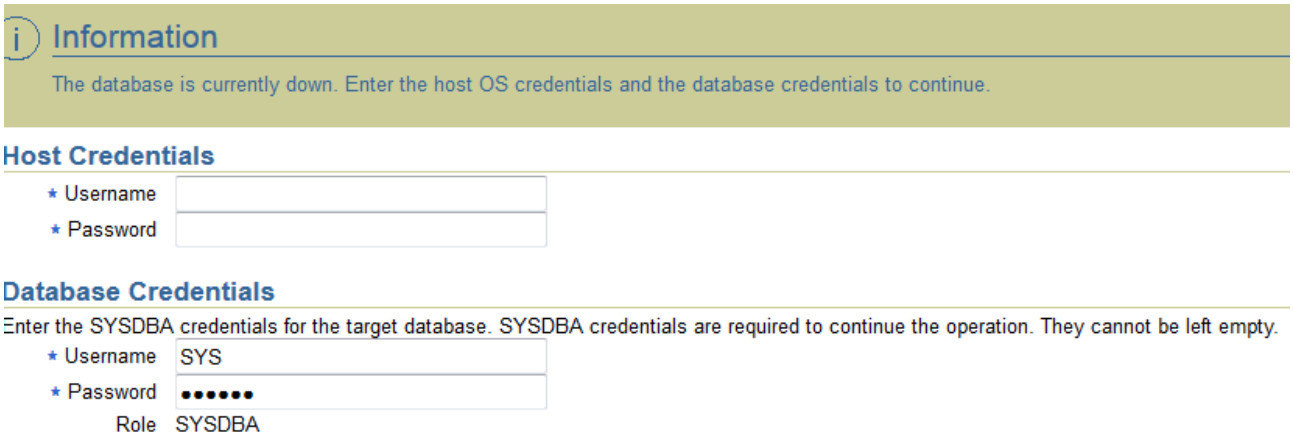
-Force a log switch and verify the switch has taken place. Perform several inserts into the HR.DEPARTMENTS2 table and commit the changes. Verify the new row count. Then exit your SQL*Plus session.

```
SQL> alter system switch logfile;

System altered.

Elapsed: 00:00:05.13
SQL> select sequence#,status from v$log;

 SEQUENCE# STATUS
---------- ----------------
         1 ACTIVE
         2 CURRENT
         0 UNUSED

Elapsed: 00:00:00.00
SQL> insert into hr.departments2 values (280,'DUMMY1',",");
insert into hr.departments2 values (280,'DUMMY1',",")
                                        *
ERROR at line 1:
ORA-00947: not enough values


Elapsed: 00:00:00.01
SQL> insert into hr.departments2 values (280,(DUMMY1',",");
ERROR:
ORA-01756: quoted string not properly terminated


Elapsed: 00:00:00.01
SQL> insert into hr.departments2 values (280, 'DUMMY1','','');

1 row created.

Elapsed: 00:00:00.02
SQL> insert into hr.departments2 values (290, 'DUMMY2','','');

1 row created.

Elapsed: 00:00:00.02
SQL> insert into hr.departments2 values (300, 'DUMMY3','','');

1 row created.

Elapsed: 00:00:00.01
SQL> commit;

Commit complete.

Elapsed: 00:00:00.01
SQL> select count(*) from hr.departments2;

  COUNT(*)
----------
        30

Elapsed: 00:00:00.01
SQL>
```

**4)**

-Using Enterprise Manager while logged in as a SYSDBA user, recover the database to a point in time before the new data was introduced using the information you recorded before the inserts were performed.




**5)**

-Wait until the Operation Succeeded message is displayed, then use SQL*Plus to verify that the recovery was successful by checking the row count in the HR.DEPARTMENTS2 table.