Storage Tiering Policy for Automatic Data Optimization ( Using Heat map )

\*\*\* Only support for non-cdb

# Creating Tablespaces, Enabling Accounts, and Enabling Heat Map Tracking

* Connect to the noncdb database as sys.

sqlplus sys/oracle@localhost:1521/noncdb as sysdba

* Create two tablespaces, one called ilmtbs, and the other called low\_cost\_store. Each tablespace should have a 10M data file.

SQL> create tablespace ilmtbs datafile '+DATA' size 10m reuse  
autoextend off extent management local;

SQL> create tablespace low\_cost\_store datafile '+DATA' size 10m reuse  
autoextend off extent management local;

* Unlock the SCOTT account, grant it unlimited quota on the two tablespaces. Grant the necessary privileges to SCOTT.

SQL> alter user scott identified by tiger account unlock;

SQL> alter user scott quota unlimited on ilmtbs;

SQL> alter user scott quota unlimited on low\_cost\_store;

SQL> grant alter tablespace, select any dictionary to scott;

SQL> grant all on ts$ to scott;

SQL> grant all on dba\_segments to scott;

* Enable heat map tracking.

SQL> alter system set heat\_map=on scope=both;

# Creating and Updating a Table in the ILMTBS Tablespace

* Create the SCOTT.employee table in the ilmtbs tablespace.

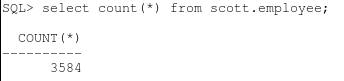
create table scott.employee (  
        EMPNO     NUMBER(4) NOT NULL,  
        ENAME     VARCHAR2(10),  
        JOB       VARCHAR2(9),  
        MGR       NUMBER(4),  
        HIREDATE  DATE,  
        SAL       NUMBER(7,2),  
        COMM      NUMBER(7,2),  
        DEPTNO    NUMBER(2)  
)  
tablespace ilmtbs;

* Insert about 3500 rows into SCOTT.employee table.

insert into scott.employee (empno, ename, job, mgr, hiredate, sal, comm, deptno)  
select empno, ename, job, mgr, hiredate, sal, comm, deptno from scott.emp  
/

declare  
blowup PLS\_INTEGER := 8;  
sql\_test clob;  
begin  
for i in 1..blowup loop  
sql\_test := 'insert /\*+ append \*/ into scott.employee select \* from scott.employee';  
execute immediate sql\_test;  
commit;  
end loop;  
end;  
/

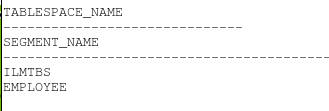
select count(\*) from scott.employee;



This should fill up more than 5% of the tablespace, so that there is less than 95% free space.

* Verify that the table is stored in the ilmtbs tablespace.

select tablespace\_name, segment\_name from dba\_segments  
where segment\_name='EMPLOYEE';



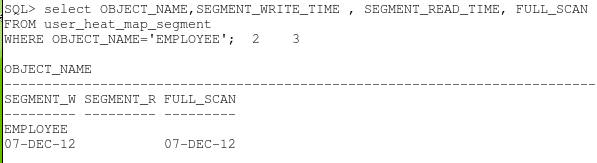
# Verifying Heat Map Tracking Collected Statistics for the Table

* Connect as scott to the database.

connect scott/tiger@localhost:1521/noncdb

* Verify that heat map tracking collected statics for SCOTT.employee.

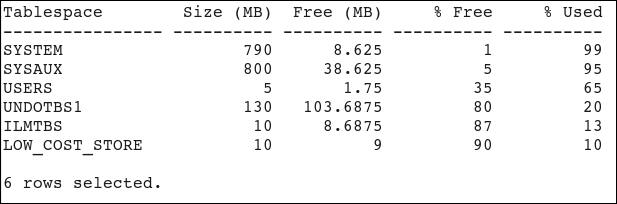
select OBJECT\_NAME,SEGMENT\_WRITE\_TIME , SEGMENT\_READ\_TIME, FULL\_SCAN  
FROM user\_heat\_map\_segment  
WHERE OBJECT\_NAME='EMPLOYEE';



* Check the current freespace in the ilmtbs tablespace.

col tablespace format A16

SELECT /\* + RULE \*/ df.tablespace\_name "Tablespace",  
df.bytes / (1024 \* 1024) "Size (MB)",  
SUM(fs.bytes) / (1024 \* 1024) "Free (MB)",  
Nvl(Round(SUM(fs.bytes) \* 100 / df.bytes),1) "% Free",  
Round((df.bytes - SUM(fs.bytes)) \* 100 / df.bytes) "% Used"  
FROM dba\_free\_space fs,  
(SELECT tablespace\_name,SUM(bytes) bytes  
FROM dba\_data\_files  
GROUP BY tablespace\_name) df  
WHERE fs.tablespace\_name (+) = df.tablespace\_name  
GROUP BY df.tablespace\_name,df.bytes  
Order by 4;



Notice that %Free is already less than 95%.

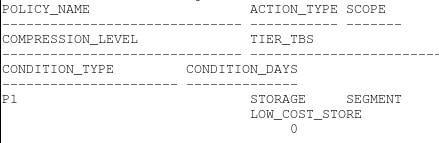
# Creating a Storage Tiering Policy on the Table

* Create a storage tiering policy on SCOTT.employee.

alter table scott.employee ilm add policy tier to low\_cost\_store;

* If heat map tracking was not enabled, you will receive an error message when you try to create the policy.
* Verify that the policy is added.

select  cast(policy\_name as varchar2(30)) policy\_name,  
  action\_type, scope, compression\_level, cast(tier\_tablespace as   
  varchar2(30)) tier\_tbs, condition\_type, condition\_days  
from  user\_ilmdatamovementpolicies  
order by policy\_name;



select \* from user\_ilmobjects;

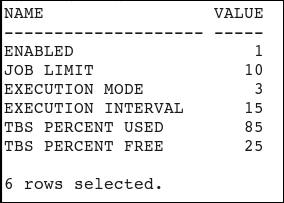
http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/12c/r1/ilm/ilm_tiering/images/t40102.jpg

# Triggering the Table to Move to Low Cost Storage

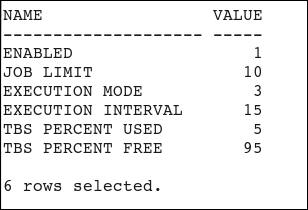
* The ILM decision to move segments also depends on the default thresholds defined at the database level for all user-defined tablespaces. Set the TBS\_PERCENT\_FREE threshold  to 95% and the TBS\_PERCENT\_USED threshold to 5%.

connect sys/oracle@localhost:1521/noncdb as sysdba  
  
col name format A20  
col value format 9999

select \* from dba\_ilmparameters;



EXEC dbms\_ilm\_admin.customize\_ilm(DBMS\_ILM\_ADMIN.TBS\_PERCENT\_FREE,95)  
  
EXEC dbms\_ilm\_admin.customize\_ilm(DBMS\_ILM\_ADMIN.TBS\_PERCENT\_USED,5)  
  
select \* from dba\_ilmparameters;



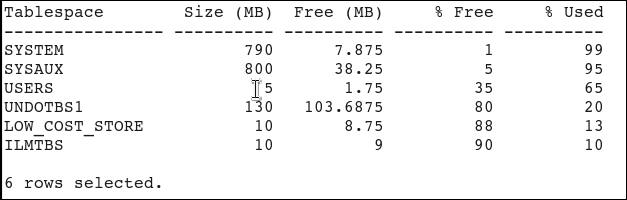
* For the purposes of this tutorial, we cannot wait for the maintenance window to open that will trigger the automatic data optimization policies jobs. Instead, you are going to use the following PL/SQL block and trigger it as the table owner.

connect scott/tiger@localhost:1521/noncdb

declare  
v\_executionid number;  
begin  
dbms\_ilm.execute\_ILM (ILM\_SCOPE => dbms\_ilm.SCOPE\_SCHEMA,  
            execution\_mode => dbms\_ilm.ilm\_execution\_offline,  
            task\_id   => v\_executionid);  
end;  
/

* Check the current free space in ILMTBS tablespace. The LOW\_COST\_STORE may show a value for the column % Used, although the space used in ILMTBS may not have decreased. If this is the case, a few seconds later, run the same statement and you will see that the data dictionary has been updated to reflect the new situation.

SELECT /\* + RULE \*/ df.tablespace\_name "Tablespace",  
df.bytes / (1024 \* 1024) "Size (MB)",  
SUM(fs.bytes) / (1024 \* 1024) "Free (MB)",  
Nvl(Round(SUM(fs.bytes) \* 100 / df.bytes),1) "% Free",  
Round((df.bytes - SUM(fs.bytes)) \* 100 / df.bytes) "% Used"  
FROM dba\_free\_space fs,  
(SELECT tablespace\_name,SUM(bytes) bytes  
FROM dba\_data\_files  
GROUP BY tablespace\_name) df  
WHERE fs.tablespace\_name (+) = df.tablespace\_name  
GROUP BY df.tablespace\_name,df.bytes  
Order by 4;



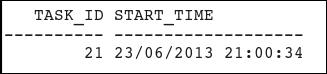
* Verify the SCOTT.employee segment was moved to the low\_cost\_store tablespace.

select tablespace\_name, segment\_name  
from dba\_segments  
where segment\_name='EMPLOYEE';

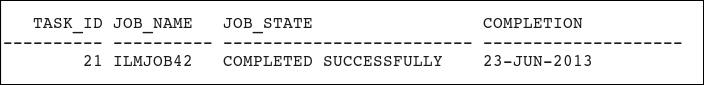


* View the results of the job that completed the movement operation.

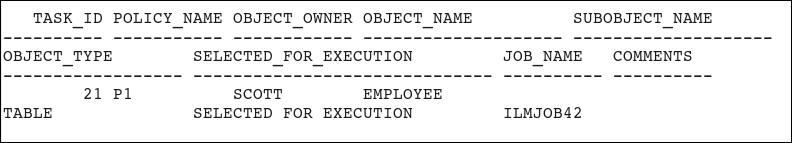
SELECT task\_id, to\_char(start\_time, 'dd/mm/yyyy hh24:mi:ss') as start\_time  
FROM user\_ilmtasks;



select task\_id, job\_name, job\_state, to\_char(completion\_time,'dd-MON-yyyy')completion  
from user\_ilmresults;



SELECT \* FROM user\_ilmevaluationdetails;



# Resetting Your Environment

* Perform the following steps to reset your environment prior to repeating the activities covered in this OBE or starting another OBE.
* Connect to the noncdb instance as sys.

connect sys/oracle@localhost:1521/noncdb as sysdba

* Delete the ILM internal tables. This is not recommended practice in production environments.

delete ilm$;  
delete ilmpolicy$;  
delete ilmobj$;  
delete ilm\_results$;  
delete ilm\_execution$;  
delete ilm\_executiondetails$;

* Drop the ilmtbs and low\_cost\_store tablespaces.

drop tablespace ilmtbs including contents and datafiles;  
  
drop tablespace low\_cost\_store including contents and datafiles;

* Reset the tablespace %Free and %Used thresholds.

exec dbms\_ilm\_admin.customize\_ilm(dbms\_ilm\_admin.tbs\_percent\_used,85);

exec dbms\_ilm\_admin.customize\_ilm(dbms\_ilm\_admin.tbs\_percent\_free,25);

* Disable heatmap tracking.

alter system set heat\_map=off scope=both;