***SESSION 7 MANUAL ALLOCATION/DEALLOCATION of TABLE EXTENTS***

Last login: Sun Feb 20 10:04:03 2022 from 10.31.12.209

[student@oracledb19c ~]$ **su - oracle**

Password:

Last login: Sun Feb 20 10:04:11 EST 2022 on pts/0

The Oracle base remains unchanged with value /opt/oracle/app/oracle

[oracle@oracledb19c ~]$ **pwd**

/home/oracle

[oracle@oracledb19c ~]$ **sqlplus / as sysdba**

SQL\*Plus: Release 19.0.0.0.0 - Production on Tue Feb 22 12:04:06 2022

Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production

Version 19.3.0.0.0

SQL> **set pagesize 120**

SQL> **set linesize 120**

SQL>

SQL> **SELECT \* FROM v$TABLESPACE;**

TS# NAME INC BIG FLA ENC CON\_ID

---------- ------------------------------ --- --- --- --- ----------

1 SYSAUX YES NO YES 0

0 SYSTEM YES NO YES 0

2 UNDOTBS1 YES NO YES 0

4 USERS YES NO YES 0

3 TEMP NO NO YES 0

6 MINE YES NO YES 0

7 JOKE YES NO YES 0

7 rows selected.

**\* For this exercise you will need to have Tablespaces MINE, JOKE and user TOM created \***

**\* If NOT then create Tablespace MINE as UNIFORM of 512K, also JOKE as UNIFORM of 80K and then create new user TOM with password “cat”, Default Tablespace MINE and Temporary Tablespace TEMP. He will be granted CONNECT role and also CREATE TABLE System Privilege, plus UNLIMITED TABLESPACE System privilege. \***

SQL> **CREATE USER TOM IDENTIFIED BY cat**

**DEFAULT TABLESPACE MINE**

**TEMPORARY TABLESPACE TEMP;**

User created.

SQL> **GRANT connect, create table, UNLIMITED TABLESPACE TO tom;**

Grant succeeded.

SQL> **SELECT name, bytes, blocks FROM v$datafile**

**WHERE ts# =6;**

NAME

---------------------------------------------------

BYTES BLOCKS

----------------------------

/opt/oracle/app/oracle/oradata/STUDENT/mine01.dbf

**10485760 1280**

SQL> **SELECT name, bytes, blocks FROM v$datafile**

**WHERE ts# =7;**

NAME

---------------------------------------------------

BYTES BLOCKS

----------------------------

/opt/oracle/app/oracle/oradata/STUDENT/joke01.dbf

**15728640 1920**

SQL> **SELECT initial\_extent, next\_extent, min\_extents FROM dba\_tablespaces**

**WHERE tablespace\_name ='MINE';**

INITIAL\_EXTENT NEXT\_EXTENT MIN\_EXTENTS

------------------------------------------

524288 524288 1

SQL> **SELECT initial\_extent, next\_extent, min\_extents FROM dba\_tablespaces**

**WHERE tablespace\_name ='JOKE';**

INITIAL\_EXTENT NEXT\_EXTENT MIN\_EXTENTS

------------------------------------------

81920 81920 1

SQL> REM 524288 bytes= 512K/8K=64 BLOCKS PER EACH EXTENT in MINE

SQL> REM 81920 bytes= 80k/8k=10 BLOCKS PER EACH EXTENT in JOKE

SQL> **SELECT tablespace\_name, allocation\_type, initial\_extent**

**FROM dba\_tablespaces ;**

TABLESPACE\_NAME ALLOCATIO INITIAL\_EXTENT

------------------------------ --------- --------------

SYSTEM SYSTEM 65536

SYSAUX SYSTEM 65536

UNDOTBS1 SYSTEM 65536

TEMP UNIFORM 1048576

**USERS SYSTEM 65536** ** 64k or 8 blocks**

MINE UNIFORM 524288

**JOKE UNIFORM 81920** ** 80k or 10 blocks**

7 rows selected.

SQL> **conn tom/cat**

Connected.

SQL> **SELECT TNAME FROM TAB;**

no rows selected

**\* Let’s create Two tables as Tom with different Storage parameters, both in Tablespace Joke**

**that was created as Uniform with 80K extents (= 10 Blocks) \***

SQL> **CREATE TABLE new\_emp( empno NUMBER(4), ename VARCHAR2(30),**

**job VARCHAR2(9), mgr NUMBER(4), hiredate DATE,**

**sal NUMBER(7,2), comm NUMBER(7,2), deptno NUMBER(2))**

**TABLESPACE JOKE STORAGE (MINEXTENTS 6);**

Table created.

SQL> **CREATE TABLE big\_emp( empno NUMBER(4), ename VARCHAR2(30))**

**TABLESPACE JOKE STORAGE (INITIAL 1M);**

Table created.

SQL> **SELECT tname FROM tab;**

TNAME

-------------------------------------------

BIG\_EMP

NEW\_EMP

SQL> **conn / as sysdba**

Connected.

SQL> **DESC dba\_segments**

Name Null? Type

----------------------------------------------------------------- -------- -

OWNER VARCHAR2(128)

SEGMENT\_NAME VARCHAR2(128)

PARTITION\_NAME VARCHAR2(128)

SEGMENT\_TYPE VARCHAR2(18)

SEGMENT\_SUBTYPE VARCHAR2(10)

TABLESPACE\_NAME VARCHAR2(30)

HEADER\_FILE NUMBER

HEADER\_BLOCK NUMBER

BYTES NUMBER

BLOCKS NUMBER

EXTENTS NUMBER

INITIAL\_EXTENT NUMBER

NEXT\_EXTENT NUMBER

MIN\_EXTENTS NUMBER

MAX\_EXTENTS NUMBER etc …

SQL> **SELECT segment\_name, segment\_type, tablespace\_name,**

**extents, blocks**

**FROM dba\_segments**

**WHERE owner = 'SCOTT';**

SEGMENT\_NAME

-----------------------------------------------------------------------------

SEGMENT\_TYPE TABLESPACE\_NAME EXTENTS BLOCKS

------------------ ------------------------------ ---------- ----------

DEPT

TABLE USERS 1 8

EMP

TABLE USERS 1 8

SALGRADE

TABLE USERS 1 8

DUMMY

TABLE USERS 1 8

CUSTOMER

TABLE USERS 1 8

ORD

TABLE USERS 1 8

DEPT\_PRIMARY\_KEY

INDEX USERS 1 8

EMP\_PRIMARY\_KEY

INDEX USERS 1 8

CUSTOMER\_PRIMARY\_KEY

INDEX USERS 1 8

ORD\_PRIMARY\_KEY

INDEX USERS 1 8

10 rows selected.

**\* Tablespace USERS is AUTOALLOCATED by System, that means it will be firstly 8 blocks per Extent and later may be more (multiples of 64K or 8 blocks). \***

SQL> **SELECT segment\_name, segment\_type, tablespace\_name,**

**extents, blocks**

**FROM dba\_segments**

**WHERE owner = 'TOM';**

no rows selected ** Since Oracle11g , FIRST EXTENT is NOT allocated when you**

**create a table, but when you insert FIRST ROW into it.**

SQL> **INSERT INTO tom.new\_emp VALUES**

**(501,'JONES',NULL,NULL, SYSDATE, 5000, NULL, NULL);**

1 row created.

SQL> **INSERT INTO tom.big\_emp VALUES (901,'HONG');**

1 row created.

SQL> **commit;**

Commit complete.

SQL> **SELECT segment\_name, segment\_type, tablespace\_name,**

**extents, blocks**

**FROM dba\_segments**

**WHERE owner = 'TOM';**

SEGMENT\_NAME

-----------------------------------------------------------------------------

SEGMENT\_TYPE TABLESPACE\_NAME EXTENTS BLOCKS

------------------ ------------------------------ ---------- ----------

NEW\_EMP

TABLE JOKE 6 60

BIG\_EMP

TABLE JOKE 13 130

**\* Let’s explain how the extents were allocated for these 2 tables for user TOM:**

**NEW\_EMP: It was created with the STORAGE clause (MINEXTENTS=6) in the tablespace JOKE**

**where all extents have UNIFORM size of 10 blocks  so it allocates 6 extents of 10 blocks  *60 blocks* total**

**BIG\_EMP: It was created with the STORAGE clause (INTIAL=1M and MINEXTENTS not specified  1 as default). So, we need here 1M only for the first big extent and that means**

**1M= 1024K/8K = *128 blocks*, but in the tablespace JOKE  so it allocates 13 Extents of 10 blocks each  *130 blocks* total \***

SQL> rem INITIAL=1M --> 128 Blocks = 12.8 extents --> we get 13 extents

**\* We can see detailed (not cumulative) Extent situation for each segment by checking dba\_extents view.\***

SQL> **DESC dba\_extents**

Name Null? Type

----------------------------------------------------------------- -------- -

OWNER VARCHAR2(128)

SEGMENT\_NAME VARCHAR2(128)

PARTITION\_NAME VARCHAR2(128)

SEGMENT\_TYPE VARCHAR2(18)

TABLESPACE\_NAME VARCHAR2(30)

EXTENT\_ID NUMBER

FILE\_ID NUMBER

BLOCK\_ID NUMBER

BYTES NUMBER

BLOCKS NUMBER

RELATIVE\_FNO NUMBER

SQL> **SELECT file\_id, extent\_id, block\_id, blocks**

**FROM dba\_extents**

**WHERE owner = 'TOM' AND segment\_name = 'NEW\_EMP';**

FILE\_ID EXTENT\_ID BLOCK\_ID BLOCKS

---------- ---------- ---------- ----------

2 0 8 10

8 1 8 10

2 2 18 10

8 3 18 10

2 4 28 10

8 5 28 10

6 rows selected.

SQL> **ALTER TABLE tom.new\_emp ALLOCATE EXTENT;**

Table altered.

SQL> **SELECT file\_id, extent\_id, block\_id, blocks**

**FROM dba\_extents**

**WHERE owner = 'TOM' AND segment\_name = 'NEW\_EMP';**

FILE\_ID EXTENT\_ID BLOCK\_ID BLOCKS

---------- ---------- ---------- ----------

2 0 8 10

8 1 8 10

2 2 18 10

8 3 18 10

2 4 28 10

8 5 28 10

2 6 108 10

7 rows selected.

**\* When we manually add Extent (without SIZE option), then it will be just another Extent of the Uniform size in the Tablespace JOKE (10 Blocks here), BLOCK\_ID points o the LEADING block of the adjacent group of Blocks in the Extent \***

SQL> **ALTER TABLE tom.new\_emp ALLOCATE EXTENT (SIZE 304K);**

Table altered.

SQL> **SELECT file\_id, extent\_id, block\_id, blocks**

**FROM dba\_extents**

**WHERE owner = 'TOM' AND segment\_name = 'NEW\_EMP';**

FILE\_ID EXTENT\_ID BLOCK\_ID BLOCKS

---------- ---------- ---------- ----------

2 0 8 10

8 1 8 10

2 2 18 10

8 3 18 10

2 4 28 10

8 5 28 10

2 6 108 10

8 7 98 10

2 8 118 10

8 9 108 10

2 10 128 10

11 rows selected.

**\* When we manually add Extent (with SIZE option), then that value will be calculated against the Uniform size value, like shown here:**

**Asked for 304k = 38 Blocks, but in tablespace JOKE all extents have UNIFORM size of 10 blocks  so, 4 Extents of 10 blocks were added  4\*10=40 *blocks* were added.**

**Next syntax shows us that we have now 11 Extents and 110 blocks for table NEW\_EMP and we used to have 7 Extents and 70 blocks \***

SQL> **SELECT file\_id, COUNT(extent\_id), SUM(blocks)**

**FROM dba\_extents**

**WHERE owner = 'TOM'**

**AND segment\_name = 'NEW\_EMP'**

**GROUP BY file\_id;**

FILE\_ID COUNT(EXTENT\_ID) SUM(BLOCKS)

---------- ---------------- -----------

2 6 60

8 5 50

SQL> **DESC dba\_tables**

Name Null? Type

----------------------------------------------------------------- -------- -

OWNER NOT NULL VARCHAR2(128)

TABLE\_NAME NOT NULL VARCHAR2(128)

TABLESPACE\_NAME VARCHAR2(30)

CLUSTER\_NAME VARCHAR2(128)

IOT\_NAME VARCHAR2(128)

STATUS VARCHAR2(8)

PCT\_FREE NUMBER

PCT\_USED NUMBER

INI\_TRANS NUMBER

MAX\_TRANS NUMBER

INITIAL\_EXTENT NUMBER

NEXT\_EXTENT NUMBER

MIN\_EXTENTS NUMBER

MAX\_EXTENTS NUMBER

PCT\_INCREASE NUMBER

FREELISTS NUMBER

FREELIST\_GROUPS NUMBER

LOGGING VARCHAR2(3)

BACKED\_UP VARCHAR2(1)

NUM\_ROWS NUMBER

BLOCKS NUMBER

EMPTY\_BLOCKS NUMBER

AVG\_SPACE NUMBER

CHAIN\_CNT NUMBER

AVG\_ROW\_LEN NUMBER etc ...

SQL> **ANALYZE TABLE tom.new\_emp COMPUTE STATISTICS;**

Table analyzed.

**\* In order to see Block situation for the High Water Mark and Above HWM we must ANALYZE table firstly with either COMPUTE option (this statistics is precise, because ALL rows were analyzed) or ESTIMATE option (it is based on sample of 1024 rows) \***

SQL> **SELECT num\_rows, blocks HWM, empty\_blocks "Above HWM"**

**FROM dba\_tables**

**WHERE owner = 'TOM' AND table\_name ='NEW\_EMP';**

NUM\_ROWS HWM Above HWM

---------- ---------- ----------

1 7 103

SQL> **ALTER TABLE tom.new\_emp DEALLOCATE UNUSED**;

Table altered.

SQL> **ANALYZE TABLE tom.new\_emp COMPUTE STATISTICS;**

Table analyzed.

SQL> **SELECT num\_rows, blocks HWM, empty\_blocks "Above HWM"**

**FROM dba\_tables**

**WHERE owner = 'TOM' AND table\_name ='NEW\_EMP';**

NUM\_ROWS HWM Above HWM

---------- ---------- ----------

1 7 53

**\* Why do we still have lots of blocks Above HWM, after using DEALLOCATE syntax?**

**Well, table NEW\_EMP was created by using STORAGE clause with MINEXTENTS 6, that always guarantees 6 Extents for this table  or calculated against the JOKE tablespace and its Uniform size of 80K (like shown above), that means 6 extents are safe (60 blocks is the minimum)  7 + 53 = 60 \***

SQL> **SELECT file\_id, COUNT(extent\_id), SUM(blocks)**

**FROM dba\_extents**

**WHERE owner = 'TOM'**

**AND segment\_name ='BIG\_EMP'**

**GROUP BY file\_id;**

FILE\_ID COUNT(EXTENT\_ID) SUM(BLOCKS)

---------- ---------------- -----------

2 7 70

8 6 60

SQL> **ALTER TABLE tom.big\_emp ALLOCATE EXTENT (SIZE 211K);**

Table altered.

**\* Asked for 211k = almost 27 Blocks, but in tablespace JOKE all extents have UNIFORM size of 10 blocks, so 3 Extents of 10 blocks were added 3\*10=30 *blocks* were added.**

**Next syntax shows us that we have now 16 Extents and 160 blocks for table BIG\_EMP and we used to have 13 Extents and 130 blocks \***

SQL> **SELECT file\_id, COUNT(extent\_id), SUM(blocks)**

**FROM dba\_extents**

**WHERE owner = 'TOM'**

**AND segment\_name = 'BIG\_EMP'**

**GROUP BY file\_id;**

FILE\_ID COUNT(EXTENT\_ID) SUM(BLOCKS)

---------- ---------------- -----------

2 8 80

8 8 80

SQL> **ANALYZE TABLE tom.BIG\_emp COMPUTE STATISTICS;**

Table analyzed.

SQL> **SELECT num\_rows, blocks HWM, empty\_blocks "Above HWM"**

**FROM dba\_tables**

**WHERE owner = 'TOM' AND table\_name = 'BIG\_EMP';**

NUM\_ROWS HWM Above HWM

---------- ---------- ----------

1 7 153

SQL> **ALTER TABLE tom.BIG\_emp DEALLOCATE UNUSED;**

Table altered.

SQL> **ANALYZE TABLE tom.BIG\_emp COMPUTE STATISTICS;**

Table analyzed.

SQL> **SELECT num\_rows, blocks HWM, empty\_blocks "Above HWM"**

**FROM dba\_tables**

**WHERE owner = 'TOM' AND table\_name = 'BIG\_EMP';**

NUM\_ROWS HWM Above HWM

---------- ---------- ----------

1 7 123

**\* Why do we still have lots of blocks Above HWM, after using DEALLOCATE syntax?**

**Well, table BIG\_EMP was created by using STORAGE clause with INITIAL=1M, that always guarantees at least 1024K/8K=128 Blocks for this table  or calculated against the JOKE tablespace and its Uniform size of 80K, that means 13 extents are safe (130 blocks is the minimum)  7 + 123 = 130 \***

SQL> **exit**