SESSION 8 MANUAL ALLOCATION of TABLE EXTENTS

```
 The Oracle base remains unchanged with value /opt/oracle
[oracle@oracloud12c ~]$ pwd
/home/oracle
[oracle@oracloud12c ~]$ cd /opt/oracle/admin/student/pfile
[oracle@oracloud12c pfile]$ ls -l
total 8
-rw-r-----. 1 oracle dba 1767 Jul 24  2017 init.ora.6242017113352
-rw-r-----. 1 oracle dba 1811 Jan 31 17:52 initstudent.ora
[oracle@oracloud12c pfile]$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Tue Feb 20 10:41:20 2018

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

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

SQL> set pagesize 120
SQL> set linesize 120
```

*** Let's create new user TOM with password "cat", Default Tablespace MINE and Temporary Tablespace TEMP. He will be granted CONNECT role (by default in EM) and also CREATE TABLE System Privilege. We will also put Quotas on 2 tablespaces for TOM ***

```
SQL> CREATE USER TOM IDENTIFIED BY cat
     DEFAULT TABLESPACE MINE
     TEMPORARY TABLESPACE TEMP;

User created.

SQL> GRANT CONNECT, CREATE TABLE TO TOM;

Grant succeeded.

SQL> ALTER  USER tom QUOTA 2M ON mine;

User altered.

SQL> ALTER  USER tom QUOTA UNLIMITED ON joke;

User altered.

SQL> CONN TOM/CAT;       ← password is Case SEnsiTIve (since 11g)
ERROR:
ORA-01017: invalid username/password; logon denied
```

```
SQL> CONN TOM/cat;
Connected.

SQL> SELECT * 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 (INITIAL 100K   NEXT 100K
                PCTINCREASE 0   MINEXTENTS 6    MAXEXTENTS 10);

Table created.

SQL> CREATE TABLE big_emp( empno NUMBER(4), ename VARCHAR2(30))
TABLESPACE JOKE STORAGE (INITIAL 1M    NEXT 1M  MAXEXTENTS 10) ;

Table created.

SQL> SELECT * FROM TAB;

TNAME
----------------------------------------------------------------------
TABTYPE  CLUSTERID
------- ----------
BIG_EMP
TABLE

EMP
TABLE

SQL> conn / as sysdba
Connected.

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
MGMT_ECM_DEPOT_TS              SYSTEM             65536
MGMT_TABLESPACE                SYSTEM             65536
```

```
MGMT_AD4J_TS                   SYSTEM              65536
MINE                           UNIFORM            524288
JOKE                           UNIFORM             81920  ← 80k or 10 blocks
MYUNDO                         SYSTEM              65536
MYTEMP                         UNIFORM           1048576

12 rows selected.

SQL> DESC DBA_SEGMENTS
 Name
Null?    Type
 ----------------------------------------------------------------- ---
 OWNER
VARCHAR2(128)
 SEGMENT_NAME
VARCHAR2(128)
 PARTITION_NAME
VARCHAR2(128)
 SEGMENT_TYPE
VARCHAR2(18)
    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
------------------ ------------------ ---------- ---------
PK_EMP
INDEX              USERS                     13        104

PK_DEPT
INDEX              USERS                      1          8

SALGRADE
TABLE              USERS                      1          8

EMP
TABLE              USERS                     17        256

DEPT
TABLE              USERS                      1          8
```

*** 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). Here, for table EMP is situation like 16*8 + 1*128 = 256 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> desc tom.new_emp
 Name
Null?    Type
 ----------------------------------------------------------------- ---
 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)

SQL> INSERT INTO tom.new_emp VALUES
              (501,'JONES',NULL,NULL, SYSDATE, 5000, NULL, NULL);

1 row created.

SQL> DESC tom.big_emp
 Name
Null?    Type
 ----------------------------------------------------------------- ---
 EMPNO
NUMBER(4)
 ENAME
VARCHAR2(30)

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
------------------ ---------------------------- ---------- ----------
BIG_EMP
TABLE              JOKE                                 13        130

NEW_EMP
TABLE              JOKE                                  8         80
```

*** Let's explain how the extents were allocated for these 2 tables for user TOM:**
EMP: It was created with the STORAGE clause (INTIAL=NEXT=100K and PCTINCREASE=0 and MINEXTENTS=6). So, we need here 6*100K=600K for the first 6 extents and that means 600K/8K = *75 blocks* is needed, but in tablespace JOKE all extents have UNIFORM size of 10 blocks → so it allocates 8 extents of 10 blocks → *80 blocks* total

BIG_EMP: It was created with the STORAGE clause (INTIAL=NEXT=1M and MINEXTENTS not specified → 1 as default). So, we need here 1M only for the first big extent and that means 1024K/8K = *128 blocks*, but in tablespace JOKE all extents have UNIFORM size of 10 blocks → so it allocates 13 Extents of 10 blocks → *130 blocks* total

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
---------- ---------- ---------- ----------
         9          0          8         10
        11          1          8         10
         9          2         18         10
        11          3         18         10
         9          4         28         10
        11          5         28         10
         9          6         38         10
        11          7         38         10

8 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
---------- ---------- ---------- ----------
         9          0          8         10
        11          1          8         10
         9          2         18         10
        11          3         18         10
         9          4         28         10
        11          5         28         10
         9          6         38         10
        11          7         38         10
         9          8        118         10

9 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.
```

*** 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 → 4 Extents of 10 blocks were added → 4*10=40 *blocks* were added.

Next syntax shows us that we have now 13 Extents and 130 blocks for table NEW_EMP and we used to have 9 Extents and 90 blocks *

```
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
---------- ---------- ---------- ----------
         9          0          8         10
        11          1          8         10
         9          2         18         10
        11          3         18         10
         9          4         28         10
        11          5         28         10
         9          6         38         10
        11          7         38         10
         9          8        118         10
        11          9        108         10
         9         10        128         10
        11         11        118         10
         9         12        138         10

13 rows selected.

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)
---------- ---------------- -----------
        11                6          60
         9                7          70

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        123

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         73
```

*** 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 JOKE tablespace and its Uniform size of 80K (like shown above) that means 8 extents are safe (80 blocks is the minimum) → 7 + 73 = 80 ***

<u>TRUNCATING TABLE</u>

```
SQL> CREATE TABLE scott.myemp AS SELECT * FROM scott.emp;

Table created.

SQL> ALTER TABLE scott.myemp ADD CONSTRAINT myemp_pk PRIMARY KEY
(empno);

Table altered.

SQL> SELECT file_id, extent_id, block_id, blocks
     FROM   dba_extents
     WHERE  owner = 'SCOTT'  AND segment_name = 'MYEMP';

   FILE_ID  EXTENT_ID   BLOCK_ID     BLOCKS
---------- ---------- ---------- ----------
         6          0        432          8

SQL> ALTER TABLE scott.myemp ALLOCATE EXTENT (SIZE 120k);
Table altered.
```

```
SQL> SELECT file_id, extent_id, block_id, blocks
     FROM   dba_extents
     WHERE  owner = 'SCOTT'  AND segment_name = 'MYEMP';


   FILE_ID  EXTENT_ID   BLOCK_ID     BLOCKS
---------- ---------- ---------- ----------
         6          0        432          8
         6          1        440          8
         6          2        448          8

SQL> ANALYZE TABLE scott.myemp ESTIMATE  STATISTICS;

Table analyzed.

SQL> SELECT num_rows, blocks HWM, empty_blocks "Above HWM"
     FROM   dba_tables
     WHERE  owner = 'SCOTT'   AND table_name ='MYEMP';

  NUM_ROWS        HWM  Above HWM
---------- ---------- ----------
        14          4         20

SQL> TRUNCATE   TABLE  scott.myemp;

Table truncated.

SQL> ANALYZE TABLE scott.myemp ESTIMATE  STATISTICS;

Table analyzed.

SQL> SELECT num_rows, blocks HWM, empty_blocks "Above HWM"
      FROM   dba_tables
      WHERE  owner = 'SCOTT'   AND   table_name ='MYEMP';

  NUM_ROWS        HWM  Above HWM
---------- ---------- ----------
         0          0          8
```

*** TRUNCATE command removes ALL rows quickly and moves HWM to the far left position and also leaves ONE EMPTY EXTENT by default (unless, if table was created with STORAGE option and MINEXTENTS parameter was used) ***

<u>REORGANIZING (MOVING) TABLE EXTENTS</u>

```
SQL> ALTER TABLE scott.myemp ALLOCATE EXTENT;

Table altered.
```

*** This one brings ONE new extent, so the total is 1+1=2 extents ***

```
SQL> ALTER TABLE scott.myemp ALLOCATE EXTENT (SIZE 160K);
Table altered.
```

```
SQL> SELECT file_id, extent_id, block_id, blocks
     FROM   dba_extents
     WHERE  owner = 'SCOTT'  AND segment_name = 'MYEMP';

   FILE_ID  EXTENT_ID   BLOCK_ID     BLOCKS
---------- ---------- ---------- ----------
         6          0        432          8
         6          1        440          8
         6          2        448          8
         6          3        456          8
         6          4        464          8
```

* This one brings THREE new extents, because table MYEMP is in Tablespace USERS, that is AUTOALLOCATED by system with Extents of 64K (8Blocks).
It was asked for 160K → 3*64K =192K > 160K, so the total is 2+3=5 extents *

* We will try to REORGANIZE the Extent situation for this table in SQL, meaning we will compress (MOVE) the Extents and hopefully reduce the # of allocated Extents.
This scenario involves 3 commands
1) MOVE the table extents
2) REBUILD the PK index
3) GATHER Fresh Statistics for the table *

```
SQL> ALTER TABLE SCOTT.MYEMP MOVE;

Table altered.

SQL> ALTER INDEX scott.myemp_pk REBUILD;

Index altered.

SQL> BEGIN DBMS_STATS.GATHER_TABLE_STATS('SCOTT', 'MYEMP'); END;
    /
PL/SQL procedure successfully completed.

SQL> SELECT file_id, extent_id, block_id, blocks
     FROM   dba_extents
      WHERE  owner = 'SCOTT'  AND  segment_name = 'MYEMP';

   FILE_ID  EXTENT_ID   BLOCK_ID     BLOCKS
---------- ---------- ---------- ----------
         6          0        480          8
```

* After doing these 3 steps, our table now has only ONE extent, and it used to have FIVE. *

<u>SHRINKING TABLE CONTENT (Reducing # of Extents)</u>

```
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
---------- ---------- ---------- ----------
         9          0          8         10
        11          1          8         10
         9          2         18         10
        11          3         18         10
         9          4         28         10
        11          5         28         10
         9          6         38         10
        11          7         38         10

8 rows selected.
```

* This shows the Extent info for user TOM and his table NEW_EMP. This table was placed in Tbsp JOKE (created with UNIFORM size of 80K → 10 Blocks) and was created with STORAGE option and parameter MINEXTENS=6. Like shown before, this will mean guaranteed 8 extents of 10 blocks each. We'll try to use option SHRINK to reduce # of extents used here. This scenario involves 2 steps:

1) Enable Row Movement for the table

2) SHRINK the table and release used space *

```
SQL> ALTER TABLE tom.new_emp ENABLE ROW MOVEMENT;

Table altered.

SQL> ALTER TABLE tom.new_emp SHRINK SPACE;

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
---------- ---------- ---------- ----------
         9          0          8         10
```

* It is clear that table TOM.NEW_EMP shrank from 8 to only 1 extent *

REORGANIZING (COALESCING) TABLESPACE FREE SPACE

```
SQL> SELECT tablespace_name, COUNT(*) "# of Free Fragments",
               SUM(bytes)/1024/1024 "Total Free Space (M)",
        MAX(bytes)/1024/1024 "Largest Free Fragment"
    FROM   dba_free_space
    GROUP BY tablespace_name
    ORDER  BY 3 DESC;

TABLESPACE_NAME                # of Free Fragments Total Free Space
(M) Largest Free Fragment
------------------------------ ------------------- -------------------
MGMT_TABLESPACE                                  1
228.625               228.625
MGMT_AD4J_TS                                     2
198.25              198.1875
SYSAUX                                          37
82.5625                 77.25
UNDOTBS1                                        39
61.375                    20
MYUNDO                                           1
45.875                45.875
MGMT_ECM_DEPOT_TS                                2
20.3125                   20
MINE                                             1
9                      9
JOKE                                             4
6.71875                4.0625
SYSTEM                                           2
4.125                     4
USERS                                            2
.5                   .3125

10 rows selected.
```

*** Let's try to Reorganize the empty space in Tablespace JOKE. This operation is much easier and quicker to perform in SQL compared to EM. We will use COALESCE option (similar to REORGANIZE=MOVE option for segments) ***

```
SQL> ALTER  TABLESPACE Joke COALESCE;

Tablespace altered.

SQL> SELECT tablespace_name, COUNT(*)  "# of Free Fragments",
              SUM(bytes)/1024/1024 "Total Free Space (M)",
              MAX(bytes)/1024/1024 "Largest Free Fragment"
     FROM   dba_free_space
     GROUP BY tablespace_name
     ORDER  BY 3 DESC;

TABLESPACE_NAME                # of Free Fragments Total Free Space
(M) Largest Free Fragment
------------------------------ ------------------- -------------------
MGMT_TABLESPACE                                  1
228.625               228.625
MGMT_AD4J_TS                                     2
198.25              198.1875
```

```
SYSAUX                                                 37
82.5625                 77.25
UNDOTBS1                                               39
61.375                    20
MYUNDO                                                  1
45.875                45.875
MGMT_ECM_DEPOT_TS                                       2
20.3125                    20
MINE                                                    1
9                      9
JOKE                                                    4
6.71875                4.0625
SYSTEM                                                  2
4.125                     4
USERS                                                   2
.5                  .3125

10 rows selected.
```

*** We see that our Coalesce option did NOT make any difference, but it usually helps. ***

```
SQL> EXIT
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```