

**Tugas : Query14**

**Nama : Darrell Weida WIhardi**

**NIM : 825240080**

---

**show user;**

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;
USER is "SYS"
SQL> █
```

**show con\_name;**

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;

CON_NAME
-----
FREEPDB1
SQL> █
```

**create tablespace ts1;**

**Penjelasan command:** Berfungsi untuk membuat tablespace baru bernama "ts1".

```
SQL> create tablespace ts1;

Tablespace created.

SQL> █
```

**SELECT DBMS\_METADATA.GET\_DDL('TABLESPACE','TS1') FROM dual;**

**Penjelasan command:** Berfungsi untuk menampilkan DDL/perintah SQL yang digunakan untuk membuat tablespace TS1. DBMS\_METADATA sebagai package, GET\_DDL sebagai fungsi, dan TABLESPACE, TS1 sebagai parameter dari tabel dummy.

```
SQL> SELECT DBMS_METADATA.GET_DDL('TABLESPACE','TS1') FROM dual;

DBMS_METADATA.GET_DDL('TABLESPACE','TS1')
-----

CREATE BIGFILE TABLESPACE "TS1" DATAFILE
SIZE 104857600
AUTOEXTEND ON NE
```

```
/*
CREATE TABLESPACE "TS1" DATAFILE
SIZE 104857600
AUTOEXTEND ON NEXT 104857600 MAXSIZE 32767M
```

```
LOGGING ONLINE PERMANENT BLOCKSIZE 8192  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE DEFAULT  
NOCOMPRESS SEGMENT SPACE MANAGEMENT AUTO;  
*/
```

**Penjelasan command:** Berfungsi untuk membuat tablespace "TS1" dengan spesifikasi lengkap (ukuran, autoextend, logging, dll).

**Penjelasan error:** Error terjadi karena tablespace 'TS1' sudah dibuat sebelumnya.

```
SQL> CREATE TABLESPACE "TS1" DATAFILE  
      SIZE 104857600  
      AUTOEXTEND ON NEXT 104857600 MAXSIZE 32767M  
      LOGGING ONLINE PERMANENT BLOCKSIZE 8192  
      EXTENT MANAGEMENT LOCAL AUTOALLOCATE DEFAULT  
      NOCOMPRESS SEGMENT SPACE MANAGEMENT AUTO;  
      2      3      4      5      6      CREATE TABLESPACE "TS1" DATAFILE  
*  
ERROR at line 1:  
ORA-01543: tablespace 'TS1' already exists  
Help: https://docs.oracle.com/error-help/db/ora-01543/
```

```
select * from dba_tablespaces  
where tablespace_name='TS1';
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi detail dari dba\_tablespaces yang dimana tablespace\_name = 'TS1'.

```
File Edit View Search Terminal Help  
LOGGING FOR EXTENT_MAN ALLOCATIO PLU SEGMENT DEF TAB_RETENTION BIG PREDICA  
-----  
ENC COMPRESS FOR DEF_INME DEF_INME DEF_INMEMORY_DI  
-----  
DEF_INMEMORY_COMP DEF_INMEMORY_SHARED DEF_INDE INDEX_COMPRES  
-----  
DEF_CELLMEMORY DEF_INMEMORY  
-----  
DEF_INMEMORY_SERVICE_NAME  
-----  
LOST_WR C  
-----  
OFF N  
  
TABLESPACE_NAME BLOCK_SIZE INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS  
-----  
MAX_EXTENTS MAX_SIZE PCT_INCREASE MIN_EXTLEN STATUS CONTENTS  
LOGGING FOR EXTENT_MAN ALLOCATIO PLU SEGMENT DEF TAB_RETENTION BIG PREDICA  
-----  
ENC COMPRESS FOR DEF_INME DEF_INME DEF_INMEMORY_DI  
-----  
DEF_INMEMORY_COMP DEF_INMEMORY_SHARED DEF_INDE INDEX_COMPRES  
-----  
DEF_CELLMEMORY DEF_INMEMORY  
-----  
DEF_INMEMORY_SERVICE_NAME  
-----  
LOST_WR C  
-----
```

--REMEMBER THE RELATION  
--TABLESPACE-SEGMENT-EXTENT-BLOCK

```
CREATE USER alex identified by alex  
default tablespace ts1;
```

**Penjelasan command:** Berfungsi untuk membuat user bernama "alex" dengan password "alex", dan menetapkan tablespace default pengguna tersebut ke "ts1".

```
SQL> CREATE USER alex identified by alex
default tablespace ts1;
2
User created.
```

**grant create session, create table, unlimited tablespace to alex;**

**Penjelasan command:** Berfungsi untuk memberikan tiga privilage (login, buat tabel, kuota tak terbatas) kepada user "alex".

```
SQL> grant create session, create table, unlimited tablespace to alex;
Grant succeeded.
```

**select \* from dba\_users**

**where username='ALEX';**

**Penjelasan command:** Berfungsi untuk menampilkan informasi detail dari dba\_users untuk user 'ALEX'.

```
USERNAME
-----
USER_ID
-----
PASSWORD
-----
ACCOUNT_STATUS      LOCK_DATE EXPIRY_DA
DEFAULT_TABLESPACE   TEMPORARY_TABLESPACE
LOCAL_TEMP_TABLESPACE CREATED
PROFILE
-----
INITIAL_RSRC_CONSUMER_GROUP
EXTERNAL_NAME
-----
PASSWORD VERSIONS E AUTHENTI P COM
LAST_LOGIN
-----
INH
-----
DEFAULT_COLLATION
-----
IMP ALL EXT PASSWORD_ MAN PRO REA DIC CLO APP MFA
```

**CREATE TABLE ALEX.EMP ( EMP\_ID NUMBER, NAME VARCHAR2(100), NOTE  
VARCHAR2(1000) );**

**Penjelasan command:** Berfungsi untuk membuat tabel bernama "EMP" di "ALEX".

```
SQL> CREATE TABLE ALEX.EMP ( EMP_ID NUMBER, NAME VARCHAR2(100), NOTE VARCHAR2(10
00) );
Table created.
```

**SELECT \* FROM DBA\_TABLES**

**WHERE TABLE\_NAME='EMP'**

**AND OWNER='ALEX';**

**Penjelasan command:** Berfungsi untuk menampilkan informasi metadata dari DBA\_TABLES untuk tabel "EMP" yang dimiliki oleh "ALEX".

```
STATUS      PCT_FREE   PCT_USED INI_TRANS MAX_TRANS INITIAL_EXTENT NEXT_EXTENT
-----      -----       -----      -----      -----      -----      -----      -----
MIN_EXTENTS MAX_EXTENTS PCT_INCREASE  FREELISTS FREELIST_GROUPS LOG_B  NUM_ROWS
-----      -----      -----      -----      -----      -----      -----      -----
BLOCKS EMPTY_BLOCKS AVG_SPACE CHAIN_CNT AVG_ROW_LEN
-----      -----      -----      -----      -----      -----
AVG_SPACE_FREELIST_BLOCKS NUM_FREELIST_BLOCKS DEGREE     INSTANCES CACHE
-----      -----      -----      -----      -----
TABLE_LO SAMPLE_SIZE LAST_ANAL PAR_IOT_TYPE      T_S_NES BUFFER_FLASH_C CELL_FL
-----      -----      -----      -----      -----      -----
ROW_MOVE GLO_USE DURATION      SKIP_COR MON
-----      -----
CLUSTER_OWNER
-----      -----
DEPENDEN COMPRESS COMPRESS_FOR      DRO REA SEG RESULT_CLU
-----      -----      -----      -----      -----
ACTIVITY_TRACKING      DML_TIMESTAMP      HAS CON INMEMORY INMEMORY
-----      -----
INMEMORY_DISTRI INMEMORY_COMPRESS INMEMORY_DUPL
-----      -----
DEFAULT_COLLATION
-----      -----
D S E E EXT HYB CELLMEMORY      CON CON EXT EXT INMEMORY_SER
-----      -----
INMEMORY_SERVICE_NAME
-----      -----
CON MEMOPTIM MEMOPTIM HAS ADM DAT LOGICAL_STA ROW HAS
-----      -----
VECTOR_INDEX_TYPE
-----      -----
```

```
SELECT * FROM DBA_SEGMENTS
WHERE OWNER='ALEX'
AND SEGMENT_NAME='EMP'; --SO ORACLE DID NOT CREATE SEGMENT, NO ROWS
IN THE TABLE
```

**Penjelasan command:** Berfungsi untuk mencari segment/lokasi ruang fisik untuk tabel "EMP" di "ALEX dengan hasil "no rows selected" yang menunjukkan bahwa segment belum dibuat karena adanya fitur Deferred Segment Creation, dimana segmen baru dibuat saat data pertama kali dimasukkan.

```
SQL> SELECT * FROM DBA_SEGMENTS
WHERE OWNER='ALEX'
AND SEGMENT_NAME='EMP';
2      3
no rows selected
```

```
INSERT INTO ALEX.EMP VALUES (1,'KHALED','the best teacher ever : ')";
```

**Penjelasan command:** Berfungsi untuk memasukkan satu baris data ke dalam tabel "ALEX.EMP".

```
SQL> INSERT INTO ALEX.EMP VALUES (1, 'KHALED', 'the best teacher ever : ');
1 row created.
```

**commit;**

**Penjelasan command:** berfungsi untuk menyimpan semua perubahan yang telah dibuat dalam database secara permanen.

```
SQL> commit;
Commit complete.
```

```
SELECT * FROM DBA_SEGMENTS
WHERE OWNER='ALEX'
AND SEGMENT_NAME='EMP';
```

**Penjelasan command:** Berfungsi untuk kembali mencari segment tabel "EMP" dengan owner "ALEX". Kali ini, setelah data dimasukkan, segmen berhasil ditemukan.

PARTITION_NAME
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME HEADER_FILE
HEADER_BLOCK BYTES BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT
MIN_EXTENTS MAX_EXTENTS MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS
FREELIST_GROUPS RELATIVE_FNO BUFFER_FLASH_C CELL_FL INMEMORY INMEMORY
INMEMORY_DISTRI INMEMORY_DUPN INMEMORY_COMPRESS CELLMEMORY
-----
OWNER
SEGMENT_NAME
PARTITION_NAME
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME HEADER_FILE
HEADER_BLOCK BYTES BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT
MIN_EXTENTS MAX_EXTENTS MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS
FREELIST_GROUPS RELATIVE_FNO BUFFER_FLASH_C CELL_FL INMEMORY INMEMORY
INMEMORY_DISTRI INMEMORY_DUPN INMEMORY_COMPRESS CELLMEMORY
-----

```
select * from dba_extents
where owner='ALEX'
and SEGMENT_NAME='EMP';
```

**Penjelasan command:** Berfungsi untuk melihat extent/potongan alokasi ruang yang membentuk segmen "EMP" dengan owner "ALEX".

SQL> select * from dba_extents where owner='ALEX' and SEGMENT_NAME='EMP'; 2 3 OWNER				
SEGMENT_NAME				
PARTITION_NAME				
SEGMENT_TYPE	TABLESPACE_NAME		EXTENT_ID	FILE_ID
BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO	-----
ALEX				
EMP				
OWNER				
SEGMENT_NAME				
PARTITION_NAME				
SEGMENT_TYPE	TABLESPACE_NAME		EXTENT_ID	FILE_ID
BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO	-----
TABLE	TS1			
776	65536	8	1024	0 21

--now let us insert 1000 rows

```
begin
for i in 1..1000
loop
```

```
insert into ALEX.EMP values (i,'just test name', 'just test comments');
end loop;
commit;
end;
```

**Penjelasan command:** blok PL/SQL yang melakukan looping sebanyak 1000 kali untuk memasukkan 1000 baris data ke tabel "ALEX.EMP", lalu melakukan commit.

```
SQL> begin
  for i in 1..1000
  loop
    insert into ALEX.EMP values (i,'just test name', 'just test comments');
  end loop;
  commit;
end;
2      3      4      5      6      7      8
9
10
11
12
13  /
```

PL/SQL procedure successfully completed.

```
SELECT * FROM DBA_SEGMENTS
WHERE OWNER='ALEX'
AND SEGMENT_NAME='EMP';
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi segmen "EMP" dengan owner "ALEX" setelah penambahan 1000 baris data.

```
-----+
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS
-----+
FREELIST_GROUPS RELATIVE_FNO BUFFER_  FLASH_C CELL_FL INMEMORY INMEMORY
-----+
INMEMORY_DISTRI INMEMORY_DUPN INMEMORY_COMPRESS CELLMEMORY
-----+
```

  

```
OWNER
```

  

```
SEGMENT_NAME
```

  

```
PARTITION_NAME
```

  

```
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          HEADER_FILE
-----+
HEADER_BLOCK      BYTES      BLOCKS      EXTENTS INITIAL_EXTENT NEXT_EXTENT
-----+
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS
-----+
FREELIST_GROUPS RELATIVE_FNO BUFFER_  FLASH_C CELL_FL INMEMORY INMEMORY
-----+
INMEMORY_DISTRI INMEMORY_DUPN INMEMORY_COMPRESS CELLMEMORY
-----+
```

  

```
SQL> ■
```

```
select * from dba_extents
where owner='ALEX'
and SEGMENT_NAME='EMP';
```

**Penjelasan command:** Berfungsi untuk melihat extent segmen "EMP" dengan owner "ALEX" setelah penambahan 1000 baris.

SEGMENT_TYPE	TABLESPACE_NAME	EXTENT_ID	FILE_ID
BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO
TABLE	TS1	0	21
776 65536 8 1024			
OWNER			
SEGMENT_NAME			
PARTITION_NAME			
SEGMENT_TYPE	TABLESPACE_NAME	EXTENT_ID	FILE_ID
BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO
ALEX			
EMP			
OWNER			
SEGMENT_NAME			
PARTITION_NAME			
SEGMENT_TYPE	TABLESPACE_NAME	EXTENT_ID	FILE_ID
BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO
TABLE	TS1	1	21
784 65536 8 1024			

--now let us insert 1000000 rows

```
begin
for i in 1..1000000
loop
insert into ALEX.EMP values (i,'just test name', 'just test comments');
end loop;
commit;
end;
```

**Penjelasan command:** blok PL/SQL yang melakukan *looping* sebanyak 1.000.000 kali untuk memasukkan 1.000.000 baris data lagi ke tabel "ALEX.EMP".

```
SQL> begin
  for i in 1..1000000
  loop
    insert into ALEX.EMP values (i,'just test name', 'just test comments');
  end loop;
  commit;
end;
2   3   4   5   6   7   8  /
PL/SQL procedure successfully completed.
```

```
SELECT * FROM DBA_SEGMENTS
WHERE OWNER='ALEX'
AND SEGMENT_NAME='EMP';
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi segmen "EMP" dengan owner "ALEX" setelah penambahan 1.000.000 baris data.

```
OWNER
-----
SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          HEADER_FILE
-----           -----
HEADER_BLOCK     BYTES      BLOCKS    EXTENTS INITIAL_EXTENT NEXT_EXTENT
-----           -----
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT INCREASE  FREELISTS
-----           -----
FREELIST_GROUPS RELATIVE_FNO BUFFER_  FLASH_C CELL_FL INMEMORY INMEMORY
-----           -----
INMEMORY_DISTRI INMEMORY_DUPL INMEMORY_COMPRESS CELLMEMORY
-----           -----
```

  

```
OWNER
-----
SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          HEADER_FILE
-----           -----
HEADER_BLOCK     BYTES      BLOCKS    EXTENTS INITIAL_EXTENT NEXT_EXTENT
-----           -----
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT INCREASE  FREELISTS
-----           -----
FREELIST_GROUPS RELATIVE_FNO BUFFER_  FLASH_C CELL_FL INMEMORY INMEMORY
-----           -----
INMEMORY_DISTRI INMEMORY_DUPL INMEMORY_COMPRESS CELLMEMORY
-----           -----
```

```
select * from dba_extents
where owner='ALEX'
and SEGMENT_NAME='EMP';
```

**Penjelasan command:** Berfungsi untuk melihat *extent* segmen "EMP" dengan owner "ALEX" setelah penambahan 1.000.000 baris.

```
64 rows selected.
```

```
select rowid,emp_id,name,note
from ALEX.emp
where emp_id=1000000;
--example
--rowid OOOOOOFFBBBBBBRRR
--OOOOOO object number
--FFF Relative File Number
--BBBBBBB The data block number
--RRR The row number
```

**Penjelasan command:** Berfungsi untuk menampilkan ROWID (alamat fisik baris) dan data lainnya untuk karyawan dengan emp\_id 1000000.

```
SQL> select rowid,emp_id,name,note  
from ALEX.emp  
where emp_id=1000000;  
2 3  
ROWID          EMP_ID  
-----  
NAME  
-----  
NOTE  
-----  
AAARv3AAAAAAABvGABJ    1000000  
just test name  
just test comments
```

**delete from ALEX.EMP;**

**Penjelasan command:** Berfungsi untuk menghapus semua baris data dari tabel "ALEX.EMP".

```
SQL> delete from ALEX.EMP;  
1001001 rows deleted.
```

**commit;**

**Penjelasan command:** berfungsi untuk menyimpan semua perubahan yang telah dibuat dalam database secara permanen.

```
SQL> commit;  
Commit complete.
```

```
SELECT * FROM DBA_SEGMENTS  
WHERE OWNER='ALEX'  
AND SEGMENT_NAME='EMP';
```

**Penjelasan command:** Berfungsi untuk memeriksa segmen "EMP" dengan owner "ALEX" setelah di-DELETE.

```

OWNER
-----
SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          HEADER_FILE
-----
HEADER_BLOCK     BYTES      BLOCKS     EXTENTS INITIAL_EXTENT NEXT_EXTENT
-----
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS
-----
FREELIST_GROUPS RELATIVE_FNO BUFFER_FLASH C CELL_FL INMEMORY INMEMORY
-----
INMEMORY_DISTRI INMEMORY_DUPL INMEMORY_COMPRESS CELLMEMORY
-----
```

  

```

OWNER
-----
SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          HEADER_FILE
-----
HEADER_BLOCK     BYTES      BLOCKS     EXTENTS INITIAL_EXTENT NEXT_EXTENT
-----
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS
-----
FREELIST_GROUPS RELATIVE_FNO BUFFER_FLASH C CELL_FL INMEMORY INMEMORY
-----
INMEMORY_DISTRI INMEMORY_DUPL INMEMORY_COMPRESS CELLMEMORY
-----
```

**select \* from dba\_extents**

**where owner='ALEX'**

**and SEGMENT\_NAME='EMP';**

**Penjelasan command:** Berfungsi untuk memeriksa extent dari segmen "EMP" setelah di-DELETE.

```
| 64 rows selected.
```

**truncate table ALEX.EMP;**

**Penjelasan command:** Berfungsi untuk menghapus semua data di tabel "ALEX.EMP" dengan metode TRUNCATE.

```

SQL> truncate table ALEX.EMP;
Table truncated.
```

**SELECT \* FROM DBA\_SEGMENTS**

**WHERE OWNER='ALEX'**

**AND SEGMENT\_NAME='EMP';**

**Penjelasan command:** Berfungsi untuk memeriksa segmen "EMP" dengan owner "ALEX" setelah operasi TRUNCATE.

```

OWNER
SEGMENT_NAME
PARTITION_NAME
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          HEADER_FILE
HEADER_BLOCK     BYTES      BLOCKS    EXTENTS INITIAL_EXTENT NEXT_EXTENT
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS
FREELIST_GROUPS RELATIVE_FNO BUFFER_ FLASH_C CELL_FL INMEMORY INMEMORY
INMEMORY_DISTRI INMEMORY_DUPL INMEMORY_COMPRESS CELLMEMORY
-----


OWNER
SEGMENT_NAME
PARTITION_NAME
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          HEADER_FILE
HEADER_BLOCK     BYTES      BLOCKS    EXTENTS INITIAL_EXTENT NEXT_EXTENT
MIN_EXTENTS MAX_EXTENTS   MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS
FREELIST_GROUPS RELATIVE_FNO BUFFER_ FLASH_C CELL_FL INMEMORY INMEMORY
INMEMORY_DISTRI INMEMORY_DUPL INMEMORY_COMPRESS CELLMEMORY
-----
```

**select \* from dba\_extents**

**where owner='ALEX'**

**and SEGMENT\_NAME='EMP';**

**Penjelasan command:** Berfungsi untuk memeriksa extent dari segmen "EMP" setelah operasi TRUNCATE.

```

SQL> select * from dba_extents
where owner='ALEX'
and  SEGMENT_NAME='EMP';
 2   3
OWNER
-----
SEGMENT_NAME
PARTITION_NAME
SEGMENT_TYPE      TABLESPACE_NAME          EXTENT_ID FILE_ID
BLOCK_ID        BYTES      BLOCKS RELATIVE_FNO
-----
```

BLOCK_ID	BYTES	BLOCKS	RELATIVE_FNO	EXTENT_ID	FILE_ID
ALEX					
EMP					

```

OWNER
SEGMENT_NAME
PARTITION_NAME
SEGMENT_TYPE      TABLESPACE_NAME          EXTENT_ID FILE_ID
BLOCK_ID        BYTES      BLOCKS RELATIVE_FNO
-----
```

TABLE	TS1	0	21
776	65536	8	1024

=====

```
show user;
```

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;
USER is "SYS"
SQL> [REDACTED]
```

```
show con_name;
```

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;
```

```
CON_NAME
-----
FREEPDB1
```

```
create table dept
```

```
(deptno NUMBER(2),
 dept_name varchar2(100 )
);
```

**Penjelasan command:** Berfungsi untuk membuat tabel baru bernama "dept" dengan dua kolom: "deptno" dan "dept\_name".

```
SQL> create table dept
(deptno NUMBER(2),
 dept_name varchar2(100 )
);
2   3   4
Table created.
```

```
select * from dba_tables
```

```
where table_name='DEPT';
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi dari DBA\_TABLES untuk tabel yang bernama 'DEPT'.

```
CLUSTER_NAME
IOT_NAME
STATUS PCT_FREE PCT_USEDINI_TRANS MAX_TRANS INITIAL_EXTENT NEXT_EXTENT
MIN_EXTENTS MAX_EXTENTS PCT_INCREASE FREELISTS FREELIST_GROUPS LOG_B NUM_ROWS
BLOCKS EMPTY_BLOCKS AVG_SPACE CHAIN_CNT AVG_ROW_LEN
AVG_SPACE FREELIST_BLOCKS NUM_FREELIST_BLOCKS DEGREE INSTANCES CACHE
TABLE LO SAMPLE_SIZE LAST_ANAL PAR_IOT_TYPE T_S_NES BUFFER_FLASH_C CELL_FL
ROW_MOVE GLO_USE DURATION SKIP_COR_MON
CLUSTER_OWNER
DEPENDENCIES COMPRESS_COMPRESS_FOR DRO REA SEG RESULT_CLU
ACTIVITY_TRACKING DML_TIMESTAMP HAS CON INMEMORY INMEMORY
INMEMORY_DISTRI INMEMORY_COMPRESS INMEMORY_DPL
DEFAULT_COLLATION
D_S S_E E EXT HYB CELLMEMORY CON CON EXT EXT INMEMORY_SER
INMEMORY_SERVICE_NAME
CON MEMOPTIM MEMOPTIM HAS ADM DAT LOGICAL_STA ROW HAS
VECTOR_INDEX_TYPE
```

```
SELECT * FROM USER_SEGMENTS
```

```
WHERE SEGMENT_NAME='DEPT';
```

**Penjelasan command:** Berfungsi untuk mencari *segment* tabel "DEPT" di user "SYS" (user aktif).

```
INMEMORY_COMPRESS CELLMEMORY
-----
DEFAULT DEFAULT DISABLED

SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
FLASH_C CELL_FL INMEMORY INMEMORY_INMEMORY_DISTRI INMEMORY_DUPL
INMEMORY_COMPRESS CELLMEMORY
-----

SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
FLASH_C CELL_FL INMEMORY INMEMORY_INMEMORY_DISTRI INMEMORY_DUPL
INMEMORY_COMPRESS CELLMEMORY
-----
```

```
INSERT into dept values ( 1, 'IT Dept' );
```

**Penjelasan command:** Berfungsi untuk memasukkan satu baris data (1, 'IT Dept') ke dalam tabel "dept".

```
SQL> INSERT into dept values ( 1, 'IT Dept' );
1 row created.
```

```
INSERT into dept values ( 2, 'FINANCE' );
```

**Penjelasan command:** Berfungsi untuk memasukkan satu baris data (2, 'FINANCE') ke dalam tabel "dept".

```
SQL> INSERT into dept values ( 2, 'FINANCE' );
1 row created.
```

```
commit;
```

**Penjelasan command:** berfungsi untuk menyimpan semua perubahan yang telah dibuat dalam database secara permanen.

```
SQL> commit;
Commit complete.
```

```
SELECT * FROM USER_SEGMENTS
```

```
WHERE SEGMENT_NAME='DEPT';
```

**Penjelasan command:** Berfungsi untuk kembali mencari *segment* tabel "DEPT".

```
INMEMORY_COMPRESS CELLMEMORY
-----
DEFAULT DEFAULT DISABLED

SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           BLOCKS   EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
-----           MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
-----           FLASH_C CELL_FL INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----           INMEMORY_COMPRESS CELLMEMORY
-----           SEGMENT_NAME
-----           PARTITION_NAME
-----           SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           BLOCKS   EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
-----           MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
-----           FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----           INMEMORY_COMPRESS CELLMEMORY
```

```
create unique index dept_inx on DEPT(deptno);
```

**Penjelasan command:** Berfungsi untuk membuat *unique index* bernama "dept\_inx" pada kolom "deptno" di tabel "DEPT".

```
SQL> create unique index dept_inx on DEPT(deptno);
Index created.
```

```
SELECT * FROM USER_SEGMENTS;
```

**Penjelasan command:** Berfungsi untuk menampilkan semua segmen yang dimiliki oleh user "SYS" (user aktif).

```
1566 rows selected.
```

```
update dept
```

```
set dept_name='XXXX';
```

**Penjelasan command:** Berfungsi untuk update kolom "dept\_name" menjadi 'XXXX' untuk semua baris di tabel "dept".

```
SQL> update dept
set dept_name='XXXX';
2
2 rows updated.
```

```
SELECT * FROM dba_SEGMENTS  
where owner='PDBTS_ADMIN';
```

**Penjelasan command:** Berfungsi untuk menampilkan semua segmen yang dimiliki oleh user 'PDBTS\_ADMIN'.

```
SQL> SELECT * FROM dba_SEGMENTS  
where owner='PDBTS_ADMIN';  
2  
no rows selected
```

```
select a.sid, a.serial#, a.username, b.used_urec, b.used_ublk  
from v$session a,  
     v$transaction b  
where a.saddr = b.ses_addr  
order by b.used_ublk desc;  
--USED_UBLK = Number of undo blocks used  
--USED_UREC = Number of undo records used
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi sesi dari v\$session dan transaksi dari v\$transaction yang sedang aktif, khususnya jumlah undo records (used\_urec) dan undo blocks (used\_ublk) yang digunakan.

```
SQL> select a.sid, a.serial#, a.username, b.used_urec, b.used_ublk  
from   v$session a,  
      v$transaction b  
where  a.saddr = b.ses_addr  
order by b.used_ublk desc;  
2      3      4      5  
      SID      SERIAL#  
-----  
USERNAME  
-----  
 USED_UREC  USED_UBLK  
-----  
          15        31640  
SYS  
          2            1
```

**ROLLBACK;**

**Penjelasan command:** Berfungsi untuk membatalkan transaksi UPDATE yang baru saja dilakukan.

```
SQL> ROLLBACK;  
|Rollback complete.
```

```
select a.sid, a.serial#, a.username, b.used_urec, b.used_ublk  
from v$session a,  
     v$transaction b
```

```
where a.saddr = b.ses_addr  
order by b.used_ublk desc;
```

**Penjelasan command:** Berfungsi untuk menampilkan kembali informasi transaksi aktif yang menggunakan undo.

```
SQL> select a.sid, a.serial#, a.username, b.used_urec, b.used_ublk  
from v$session a,  
      v$transaction b  
where a.saddr = b.ses_addr  
order by b.used_ublk desc;  
2   3   4   5  
no rows selected
```

--DEFERRED\_SEGMENT\_CREATION

```
show user;
```

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;  
USER is "SYS"  
SQL>
```

```
show con_name;
```

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;  
  
CON_NAME  
-----  
FREEPDB1
```

```
show parameter DEFERRED_SEGMENT_CREATION
```

**Penjelasan command:** Berfungsi untuk menampilkan nilai dari parameter “DEFERRED\_SEGMENT\_CREATION”.

```
SQL> show parameter DEFERRED_SEGMENT_CREATION
```

NAME	TYPE	VALUE
deferred_segment_creation	boolean	TRUE

--this mean the segment will created for the table

--1 after you crate the table using DDL statement ( create table )

--2 after you insert one record ( DML )

```
create table country (id number, name varchar2(100));
```

**Penjelasan command:** Berfungsi untuk membuat tabel baru "country" yang memiliki dua kolom, yaitu "id" (tipe data number) dan "name" (tipe data varchar2).

```
SQL> create table country (id number, name varchar2(100) );  
Table created.
```

```
select * from user_segments  
where segment_name= upper('country');
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi segmen dari view user\_segments untuk objek dengan nama "country".

```
INMEMORY_COMPRESS CELLMEMORY  
-----  
DEFAULT DEFAULT DISABLED  
  
SEGMENT_NAME  
-----  
PARTITION_NAME  
-----  
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME BYTES  
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS  
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER  
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL  
  
INMEMORY_COMPRESS CELLMEMORY  
-----  
  
SEGMENT_NAME  
-----  
PARTITION_NAME  
-----  
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME BYTES  
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS  
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER  
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL  
  
INMEMORY_COMPRESS CELLMEMORY  
-----
```

```
insert into country values (1,'Jordan');
```

**Penjelasan command:** Berfungsi untuk memasukkan satu baris data baru ke dalam tabel "country" dengan nilai "1" untuk kolom "id" dan "Jordan" untuk kolom "name".

```
SQL> insert into country values (1,'Jordan');  
1 row created.
```

```
select * from user_segments  
where segment_name= upper('country');
```

**Penjelasan command:** Berfungsi untuk memeriksa kembali informasi segmen untuk tabel "country" setelah data dimasukkan.

```
INMEMORY_COMPRESS CELLMEMORY
-----
DEFAULT DEFAULT DISABLED

SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           BLOCKS   EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
FLASH_C CELL_FL INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----
INMEMORY_COMPRESS CELLMEMORY
-----

SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           BLOCKS   EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----
INMEMORY_COMPRESS CELLMEMORY
```

--now let change DEFERRED\_SEGMENT\_CREATION to be false

--you can do this by alter session / alter system

--this will create the segment immidiate

**alter session set DEFERRED\_SEGMENT\_CREATION=false;**

**Penjelasan command:** Berfungsi untuk mengubah pengaturan parameter "DEFERRED\_SEGMENT\_CREATION" menjadi "false" hanya untuk sesi ini.

```
SQL> alter session set DEFERRED_SEGMENT_CREATION=false;
Session altered.
```

**show parameter DEFERRED\_SEGMENT\_CREATION**

**Penjelasan command:** Berfungsi untuk menampilkan nilai parameter "DEFERRED\_SEGMENT\_CREATION" untuk memeriksa perubahan yang baru dilakukan.

```
SQL> show parameter DEFERRED_SEGMENT_CREATION
NAME                      TYPE        VALUE
-----                    boolean    FALSE
deferred_segment_creation
SQL>
```

**create table jobs (id number, title varchar2(100) );**

**Penjelasan command:** Berfungsi untuk membuat tabel baru bernama "jobs" dengan kolom "id" dan "title".

```

SQL> show parameter DEFERRED_SEGMENT_CREATION

NAME                      TYPE        VALUE
-----
deferred_segment_creation      boolean     FALSE
SQL> create table jobs (id number, title varchar2(100) );

Table created.

```

```

select * from user_segments
where segment_name= upper('jobs');

```

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen untuk tabel "jobs" yang baru dibuat.

```

INMEMORY_COMPRESS CELLMEMORY
-----
DEFAULT DEFAULT DISABLED

SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
BLOCKS          RETENTI MINRETENTION PCT_INCREASE   FREELISTS FREELIST_GROUPS BUFFER_
MAX_SIZE        FLASH_C CELL_FL INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
FLASH_C          CELL_FL INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
INMEMORY_COMPRESS CELLMEMORY
-----

SEGMENT_NAME
-----
PARTITION_NAME
-----
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----           EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
BLOCKS          RETENTI MINRETENTION PCT_INCREASE   FREELISTS FREELIST_GROUPS BUFFER_
MAX_SIZE        FLASH_C CELL_FL INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
FLASH_C          CELL_FL INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
INMEMORY_COMPRESS CELLMEMORY
-----
```

--note

--Deferred segment is the default for tables, indexes, and partitions

--let do this now

```
alter session set DEFERRED_SEGMENT_CREATION=true;
```

**Penjelasan command:** Berfungsi untuk mengembalikan pengaturan parameter "DEFERRED\_SEGMENT\_CREATION" ke nilai "true"/default untuk sesi ini.

```

SQL> alter session set DEFERRED_SEGMENT_CREATION=true;

Session altered.

```

--but we can do this also

```
create table test ( n number )
SEGMENT CREATION IMMEDIATE;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "test" dan memaksa pembuatan segmen secara langsung (IMMEDIATE), mengabaikan pengaturan parameter sesi.

```
SQL> create table test ( n number )
SEGMENT CREATION IMMEDIATE;
 2  create table test ( n number )
*
ERROR at line 1:
ORA-00955: name is already used by an existing object
Help: https://docs.oracle.com/error-help/db/ora-00955/
```

```
select * from user_segments
```

```
where segment_name= upper('test');
```

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen untuk tabel "test".

```
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
INMEMORY_COMPRESS CELLMEMORY

SEGMENT_NAME
PARTITION_NAME
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME BYTES
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
INMEMORY_COMPRESS CELLMEMORY
```

```
create table test10 ( n number )
```

```
SEGMENT CREATION DEFERRED;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "test10" dan meminta pembuatan segmen ditunda (DEFERRED).

```
SQL> create table test10 ( n number )
SEGMENT CREATION DEFERRED;
 2  create table test10 ( n number )
*
ERROR at line 1:
ORA-14224: Deferred segment creation is not supported for this schema
Help: https://docs.oracle.com/error-help/db/ora-14224/
```

```
select * from user_segments
```

```
where segment_name= upper('test10');
```

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen untuk tabel "test10".

```
SQL> select * from user_segments  
where segment_name= upper('test10');  
2  
no rows selected
```

--Space-Saving Features (UNUSABLE index)

**show user;**

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;  
USER is "SYS"  
SQL> █
```

**show con\_name;**

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;  
  
CON_NAME  
-----  
FREEPDB1
```

**show parameter DEFERRED\_SEGMENT\_CREATION**

**Penjelasan command:** Berfungsi untuk menampilkan nilai parameter “DEFERRED\_SEGMENT\_CREATION” saat ini.

```
SQL> show parameter DEFERRED_SEGMENT_CREATION
```

NAME	TYPE	VALUE
deferred_segment_creation	boolean	TRUE

```
create table employees  
( emp_id number,  
  fname varchar2(100),  
  lname varchar2(100),  
  constraint employees_pk primary key (emp_id)  
);
```

**Penjelasan command:** Berfungsi untuk membuat tabel “employees” dengan tiga kolom dan dengan kolom “emp\_id” sebagai Primary Key.

```
SQL> show parameter DEFERRED_SEGMENT_CREATION
NAME                                     TYPE        VALUE
-----
deferred_segment_creation              boolean      TRUE
SQL> create table employees
( emp_id number,
  fname varchar2(100),
  lname varchar2(100),
  constraint employees_pk primary key (emp_id)
);
2   3   4   5   6
Table created.
```

```
select * from user_tables
where table_name='EMPLOYEES';
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi tentang tabel “EMPLOYEES” dari view user\_tables.

```
-----+
AVG_SPACE_FREELIST_BLOCKS NUM_FREELIST_BLOCKS DEGREE      INSTANCES CACHE
-----+
TABLE_10 SAMPLE_SIZE LAST_ANAL PAR_IOT_TYPE      T S NES BUFFER_ FLASH_C CELL_FL
-----+
ROW_MOVE GLO USE DURATION      SKIP_COR MON
-----+
CLUSTER_OWNER
-----+
DEPENDENT COMPRESS COMPRESS_FOR          DRO REA SEG RESULT_ CLU
-----+
ACTIVITY_TRACKING      DML_TIMESTAMP      HAS CON INMEMORY INMEMORY
-----+
INMEMORY_DISTRI INMEMORY_COMPRESS INMEMORY_DUPL
-----+
DEFAULT_COLLATION
-----+
D S S E E EXT HYB CELLMEMORY          CON CON EXT EXT INMEMORY SER
-----+
INMEMORY_SERVICE_NAME
-----+
CON MEMOPTIM MEMOPTIM HAS ADM DAT LOGICAL_ STA ROW HAS
-----+
VECTOR_INDEX_TYPE
-----+
```

```
SELECT * FROM USER_INDEXES;
```

**Penjelasan command:** Berfungsi untuk menampilkan semua indeks yang dimiliki oleh user saat ini.

```
2348 rows selected.
```

```
select * from user_segments
where segment_name IN ( 'EMPLOYEES','EMPLOYEES_PK' );
```

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen untuk tabel “EMPLOYEES” dan indeks “EMPLOYEES\_PK”.

```
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
----- MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_
----- FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
----- INMEMORY_COMPRESS CELLMEMORY
----- SEGMENT_NAME
----- PARTITION_NAME
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME BYTES
----- BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
----- MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_
----- FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
----- INMEMORY_COMPRESS CELLMEMORY
----- SQL>
```

**insert into employees values (1,'khaled','alkhudari');**

**Penjelasan command:** Berfungsi untuk memasukkan satu baris data ke dalam tabel “employees”.

```
SQL> insert into employees values (1,'khaled','alkhudari');

1 row created.
```

**commit;**

**Penjelasan command:** berfungsi untuk menyimpan semua perubahan yang telah dibuat dalam database secara permanen.

```
SQL> commit;

Commit complete.
```

**select \* from employees;**

**Penjelasan command:** Berfungsi untuk menampilkan semua data dari tabel “employees”.

```
SQL> select * from employees;
EMP_ID
-----
FNAME
-----
LNAME
-----
      1
    khaled
  alkhudari
```

```
select * from user_segments  
where segment_name IN ( 'EMPLOYEES','EMPLOYEES_PK' );
```

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen untuk tabel “EMPLOYEES” dan indeks “EMPLOYEES\_PK”.

```
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS  
-----  
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_  
-----  
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL  
-----  
INMEMORY_COMPRESS CELLMEMORY  
-----  
  
SEGMENT_NAME  
-----  
PARTITION_NAME  
-----  
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME BYTES  
-----  
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS  
-----  
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_  
-----  
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL  
-----  
INMEMORY_COMPRESS CELLMEMORY  
-----
```

```
create index fname_index on EMPLOYEES(fname) ;
```

**Penjelasan command:** Berfungsi untuk membuat indeks baru bernama “fname\_index” pada kolom “fname” di tabel “EMPLOYEES”.

```
SQL> create index fname_index on EMPLOYEES(fname) ;  
Index created.
```

```
select * from user_segments
```

```
where segment_name IN ( 'EMPLOYEES','EMPLOYEES_PK','FNAME_INDEX');
```

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen untuk tabel dan kedua indeksnya (“EMPLOYEES\_PK” dan “FNAME\_INDEX”).

```

-----+
BLOCKS      EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
-----+
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
-----+
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----+
INMEMORY_COMPRESS CELLMEMORY
-----+



SEGMENT_NAME
-----+
PARTITION_NAME
-----+
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----+
BLOCKS      EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
-----+
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
-----+
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----+
INMEMORY_COMPRESS CELLMEMORY
-----+

```

**create index lname\_index on EMPLOYEES(lname) UNUSABLE ;**

**Penjelasan command:** Berfungsi untuk membuat definisi indeks baru bernama “lname\_index”, tetapi dalam status “UNUSABLE” (tidak dapat digunakan) dan segmen penyimpanannya tidak dialokasikan.

```

SQL> create index lname_index on EMPLOYEES(lname) UNUSABLE ;

Index created.

```

**select \* from user\_segments**

**where segment\_name IN ( 'EMPLOYEES','EMPLOYEES\_PK'**  
`, 'FNAME_INDEX','LNAME_INDEX');`

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen dari semua objek terkait, tetapi “LNAME\_INDEX” tidak akan muncul karena dibuat sebagai “UNUSABLE”.

```

-----+
BLOCKS      EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
-----+
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
-----+
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----+
INMEMORY_COMPRESS CELLMEMORY
-----+



SEGMENT_NAME
-----+
PARTITION_NAME
-----+
SEGMENT_TYPE      SEGMENT_SU TABLESPACE_NAME          BYTES
-----+
BLOCKS      EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS
-----+
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE  FREELISTS FREELIST_GROUPS BUFFER_
-----+
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL
-----+
INMEMORY_COMPRESS CELLMEMORY
-----+

```

```
alter index LNAME_INDEX REBUILD;
```

**Penjelasan command:** Berfungsi untuk rebuild indeks "LNAME\_INDEX", sehingga membuat indeks tersebut menjadi "USABLE" dan mengalokasikan segmen penyimpanannya.

```
SQL> alter index LNAME_INDEX REBUILD;  
Index altered.
```

```
select * from user_segments  
where segment_name IN ('EMPLOYEES','EMPLOYEES_PK'  
'FNAME_INDEX','LNAME_INDEX');
```

**Penjelasan command:** Berfungsi untuk memeriksa informasi segmen, "LNAME\_INDEX" akan muncul karena telah di-rebuild dan memiliki segmen.

```
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS  
-----  
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_  
-----  
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL  
-----  
INMEMORY_COMPRESS CELLMEMORY  
-----  
  
SEGMENT_NAME  
-----  
PARTITION_NAME  
-----  
SEGMENT_TYPE SEGMENT_SU TABLESPACE_NAME BYTES  
-----  
BLOCKS EXTENTS INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS MAX_EXTENTS  
-----  
MAX_SIZE RETENTI MINRETENTION PCT_INCREASE FREELISTS FREELIST_GROUPS BUFFER_  
-----  
FLASH_C CELL_FL INMEMORY INMEMORY INMEMORY_DISTRI INMEMORY_DUPL  
-----  
INMEMORY_COMPRESS CELLMEMORY  
-----
```

--basic compression

--With basic compression, the Oracle Database server compresses data at the time of performing bulk load using

--operations such as direct loads or CREATE TABLE AS SELECT

--Is recommended for bulk loading data warehouses

```
show user;
```

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;  
USER is "SYS"  
SQL> [REDACTED]
```

```
show con_name;
```

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;

CON_NAME
-----
FREEPDB1
```

```
select * from dba_objects
where rownum<10000;
```

**Penjelasan command:** Berfungsi untuk menampilkan 9999 baris pertama dari view "dba\_objects".

```
9999 rows selected.
```

```
--case 1
--no compression
```

```
create table test_01
```

```
as
```

```
select * from dba_objects where rownum <= 10000;
```

**Penjelasan command:** Berfungsi untuk membuat tabel baru "test\_01" dengan struktur dan data yang disalin dari 10.000 baris pertama "dba\_objects".

```
SQL> create table test_01
as
select * from dba_objects where rownum <= 10000;
2    3
Table created.
```

```
SELECT DBMS_METADATA.GET_DDL('TABLE','TEST_01') FROM dual;
```

**Penjelasan command:** Berfungsi untuk menampilkan DDL/perintah SQL yang digunakan untuk membuat tabel 'TEST\_01'. DBMS\_METADATA sebagai package, GET\_DDL sebagai fungsi, dan TABLE sebagai parameter dari tabel dummy.

```
SQL> SELECT DBMS_METADATA.GET_DDL('TABLE','TEST_01') FROM dual;

DBMS_METADATA.GET_DDL('TABLE','TEST_01')
-----

CREATE TABLE "SYS"."TEST_01"
(    "OWNER" VARCHAR2(128),
    "OBJECT_NAME" VAR
```

```
ANALYZE TABLE TEST_01 COMPUTE statistics
```

```
select blocks, pct_free , compression, compress_for
from user_tables
where table_name = 'TEST_01';
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi alokasi (blocks, pct\_free) dan status kompresi (compression, compress\_for) untuk tabel TEST\_01.

```
SQL> select blocks, pct_free , compression, compress_for
  from user_tables
  where table_name = 'TEST_01';
  2   3
    BLOCKS    PCT_FREE  COMPRESS COMPRESS_FOR
  -----
          10      DISABLED
```

--CASE 2

--with basic compression enabled

```
create table test_02 compress basic
as
```

```
select * from dba_objects where rownum <= 10000;
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi alokasi dan kompresi untuk tabel TEST\_02 (memverifikasi bahwa kompresi ENABLED BASIC ).

```
SQL> create table test_02 compress basic
as
select * from dba_objects where rownum <= 10000;
  2   3
Table created.
```

```
ANALYZE TABLE TEST_02 COMPUTE statistics
```

```
select blocks, pct_free , compression, compress_for
  from user_tables
```

```
where table_name = 'TEST_02';
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi alokasi dan kompresi untuk tabel TEST\_02. Hasilnya memverifikasi kompresi ENABLED BASIC.

```
SQL> select blocks, pct_free , compression, compress_for
  from user_tables
  where table_name = 'TEST_02';
  2   3
    BLOCKS    PCT_FREE  COMPRESS COMPRESS_FOR
  -----
          0      ENABLED  BASIC
```

--CASE 3

--Normal insert into empty table defined as compress basic ( no effects )

```
create table test_03 compress basic
as
select * from dba_objects where 1=2;
```

**Penjelasan command:** Berfungsi untuk membuat tabel test\_03 dengan struktur dari dba\_objects dan mengaktifkan compress basic. Kondisi where 1=2 karena selalu salah, digunakan untuk membuat tabel kosong tanpa menyalin data.

```
SQL> create table test_03 compress basic  
as  
select * from dba_objects where 1=2;  
 2   3  
Table created.
```

**select \* from test\_03;**

**Penjelasan command:** Berfungsi untuk menampilkan data dari tabel "test\_03" (untuk konfirmasi bahwa tabel tersebut kosong).

```
SQL> select * from test_03;  
  
no rows selected
```

**insert into test\_03**

**select \* from dba\_objects where rownum <= 10000;**

**Penjelasan command:** Berfungsi untuk memasukkan 10.000 baris data ke dalam tabel "test\_03" menggunakan perintah "insert".

```
SQL> insert into test_03  
select * from dba_objects where rownum <= 10000;  
 2  
10000 rows created.
```

**commit;**

**Penjelasan command:** berfungsi untuk menyimpan semua perubahan yang telah dibuat dalam database secara permanen.

```
SQL> commit;  
  
Commit complete.
```

```
select blocks, pct_free , compression, compress_for  
from user_tables  
where table_name = 'TEST_03';
```

**Penjelasan command:** Berfungsi untuk memeriksa status kompresi dan alokasi blok untuk tabel "TEST\_03" setelah "insert".

```

SQL> select blocks, pct_free , compression, compress_for
  from user_tables
 where table_name = 'TEST_03';
 2   3
    BLOCKS      PCT_FREE  COMPRESS  COMPRESS_FOR
-----  -----
          0        ENABLED    BASIC

```

ANALYZE TABLE TEST\_03 COMPUTE statistics

```

select blocks, pct_free , compression, compress_for
from user_tables
where table_name = 'TEST_03';

```

**Penjelasan command:** Berfungsi untuk memeriksa status kompresi dan alokasi blok untuk tabel "TEST\_03" setelah "insert".

```

SQL> select blocks, pct_free , compression, compress_for
  from user_tables
 where table_name = 'TEST_03';
 2   3
    BLOCKS      PCT_FREE  COMPRESS  COMPRESS_FOR
-----  -----
          0        ENABLED    BASIC

```

--4. Direct path insert into empty table defined as compressed

```

create table TEST_04 compress basic
as

```

```
select * from dba_objects where rownum = 0;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "TEST\_04" dengan kompresi dasar diaktifkan, dan tabel dibuat kosong (kondisi rownum = 0).

```

SQL> create table TEST_04 compress basic
as
select * from dba_objects where rownum = 0;
 2   3
Table created.

```

```
select * from TEST_04;
```

**Penjelasan command:** Berfungsi untuk memastikan tabel "TEST\_04" kosong.

```

SQL> select * from TEST_04;
no rows selected

```

```

insert /*+ append */ into TEST_04 select * from dba_objects where rownum <= 10000
/*

```

```
SQL> insert into TEST_04 select * from dba_objects where rownum <= 10000;
10000 rows created.

SQL>
```

**Penjelasan command:** Berfungsi untuk memasukkan 10.000 baris data ke "TEST\_04" menggunakan mode direct-path insert (ditandai dengan hint /\*+ append \*/), dimana mode ini akan menerapkan kompresi dasar.

The APPEND hint tells the optimizer to perform a direct-path insert, which improves the performance of INSERT .. SELECT operations for a number of reasons:

Data is appended to the end of the table, rather than attempting to use existing free space within the table.

Data is written directly to the data files, by-passing the buffer cache.

\*/

```
ANALYZE TABLE TEST_04 COMPUTE statistics
```

```
select blocks, pct_free , compression, compress_for
from user_tables
where table_name = 'TEST_04';
```

**Penjelasan command:** Berfungsi untuk memeriksa status kompresi dan alokasi blok untuk tabel "TEST\_04" setelah *direct-path insert*.

```
SQL> select blocks, pct_free , compression, compress_for
  from user_tables
  where table_name = 'TEST_04';
   2   3
    BLOCKS    PCT_FREE  COMPRESSION  COMPRESS_FOR
-----  -----
          0      ENABLED      BASIC
```

--case 5

--without compression, then change to compressed

```
create table test_05
as
select * from dba_objects where rownum <= 10000;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "test\_05" (CTAS) tanpa kompresi, diisi dengan 10.000 baris data.

```
SQL> create table test_05
  as
  select * from dba_objects where rownum <= 10000;
   2   3
Table created.
```

```
select blocks, pct_free , compression, compress_for  
from user_tables  
where table_name = 'TEST_05';
```

**Penjelasan command:** Berfungsi untuk memeriksa status tabel "TEST\_05" dan memastikan kompresi "DISABLED".

```
SQL> select blocks, pct_free , compression, compress_for  
from user_tables  
where table_name = 'TEST_05';  
2 3  
BLOCKS PCT_FREE COMPRESS COMPRESS_FOR  
-----  
10 DISABLED
```

**alter table test\_05 compress basic;**

**Penjelasan command:** Berfungsi untuk mengubah properti tabel "test\_05" agar menggunakan kompresi dasar. Perintah ini hanya mengubah metadata, data yang sudah ada tidak dikompres.

```
SQL> alter table test_05 compress basic;  
  
Table altered.
```

```
select blocks, pct_free , compression, compress_for  
from user_tables  
where table_name = 'TEST_05';
```

**Penjelasan command:** Berfungsi untuk memeriksa status tabel "TEST\_05". Hasilnya akan menunjukkan kompresi "ENABLED", tetapi jumlah blok data masih sama (karena data lama belum dikompres).

```
SQL> select blocks, pct_free , compression, compress_for  
from user_tables  
where table_name = 'TEST_05';  
2 3  
BLOCKS PCT_FREE COMPRESS COMPRESS_FOR  
-----  
10 ENABLED BASIC
```

**alter table test\_05 move;**

**Penjelasan command:** Berfungsi untuk memindahkan tabel "test\_05". Saat tabel dipindahkan, data akan ditulis ulang dan properti "compress basic" yang sudah aktif akan diterapkan pada data tersebut.

```
SQL> alter table test_05 move;  
  
Table altered.
```

```
ANALYZE TABLE TEST_05 COMPUTE statistics
```

```
select blocks, pct_free , compression, compress_for  
from user_tables
```

```
where table_name = 'TEST_05';
```

**Penjelasan command:** Berfungsi untuk memeriksa status tabel "TEST\_05" setelah dipindahkan. Jumlah blok seharusnya berkurang karena data kini terkompresi.

```
SQL> select blocks, pct_free , compression, compress_for  
from user_tables  
where table_name = 'TEST_05';  
2 3  
BLOCKS    PCT_FREE COMPRESS COMPRESS_FOR  
-----  
          0  ENABLED   BASIC
```

--Advanced row compression

```
/*
```

Rows inserted with or without using direct-path insert and updated rows are compressed using advanced row compression.

Is enabled with CREATE TABLE ... ROW STORE COMPRESS ADVANCED

Is recommended for active OLTP environments

With advanced row compression, duplicate values in the rows and columns in a data block are stored once at the beginning of the block

note 1:

ROW STORE COMPRESS ADVANCED and COMPRESS BASIC are not supported for tables with more than 255

note 2: You cannot drop a column from a table that is compressed for direct-load operations, although you can set such a column as unused.

```
*/
```

```
create table test_06  
as  
select * from dba_objects where rownum <= 20000;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "test\_06" (CTAS) tanpa kompresi, diisi dengan 20.000 baris data.

```
SQL> create table test_06
as
select * from dba_objects where rownum <= 20000;
 2   3
Table created.
```

ANALYZE TABLE test\_06 COMPUTE statistics

```
select blocks, pct_free , compression, compress_for
from user_tables
where table_name = 'TEST_06';
```

**Penjelasan command:** Berfungsi untuk memeriksa status tabel "TEST\_06" (kompresi "DISABLED").

```
SQL> select blocks, pct_free , compression, compress_for
from user_tables
where table_name = 'TEST_06';

 2   3
BLOCKS    PCT_FREE  COMPRESS  COMPRESS_FOR
-----  -----
          10      DISABLED
```

**create table test\_07 ROW STORE COMPRESS ADVANCED**

as

```
select * from dba_objects where rownum <= 20000;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "test\_07" (CTAS) dengan mengaktifkan kompresi tingkat lanjut ("ROW STORE COMPRESS ADVANCED"), data akan langsung dikompres.

```
SQL> create table test_07 ROW STORE COMPRESS ADVANCED
as
select * from dba_objects where rownum <= 20000;
 2   3
Table created.
```

ANALYZE TABLE test\_07 COMPUTE statistics

```
select blocks, pct_free , compression, compress_for
from user_tables
where table_name = 'TEST_07';
```

**Penjelasan command:** Berfungsi untuk memeriksa status kompresi tabel "TEST\_07" (menunjukkan "ENABLED" dan "ADVANCED").

```
SQL> select blocks, pct_free , compression, compress_for
  from user_tables
 where table_name = 'TEST_07';
 2   3
    BLOCKS    PCT_FREE  COMPRESS  COMPRESS_FOR
-----  -----
          10  ENABLED    ADVANCED
```

--not let us try normal insert

```
create table test_08 ROW STORE COMPRESS ADVANCED
as
```

```
select * from dba_objects where rownum = 0;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "test\_08" kosong dengan kompresi tingkat lanjut diaktifkan.

```
SQL> create table test_08 ROW STORE COMPRESS ADVANCED
as
select * from dba_objects where rownum = 0;
 2   3
Table created.
```

```
select * from test_08;
```

**Penjelasan command:** Berfungsi untuk memastikan tabel "test\_08" kosong.

```
SQL> select * from test_08;
no rows selected
```

```
insert into test_08
```

```
select * from dba_objects where rownum <= 20000;
```

**Penjelasan command:** Berfungsi untuk memasukkan 20.000 baris data ke "test\_08" menggunakan "insert", kompresi "ADVANCED" akan diterapkan pada data ini.

```
SQL> insert into test_08
select * from dba_objects where rownum <= 20000;
 2
20000 rows created.
```

```
commit;
```

**Penjelasan command:** berfungsi untuk menyimpan semua perubahan yang telah dibuat dalam database secara permanen.

```
SQL> commit;
Commit complete.
```

```
ANALYZE TABLE test_08 COMPUTE statistics
```

```
select blocks, pct_free , compression, compress_for  
from user_tables
```

```
where table_name = 'TEST_08';
```

**Penjelasan command:** Berfungsi untuk memeriksa status kompresi tabel "TEST\_08" setelah "insert".

```
SQL> select blocks, pct_free , compression, compress_for  
from user_tables  
where table_name = 'TEST_08';  
2 3  
BLOCKS    PCT_FREE COMPRESS COMPRESS_FOR  
-----  
          10  ENABLED   ADVANCED
```

```
show user;
```

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;  
USER is "SYS"  
SQL>
```

```
show con_name;
```

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;  
  
CON_NAME  
-----  
FREEPDB1
```

```
CREATE TABLESPACE tbsalert
```

```
DATAFILE '/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'
```

```
SIZE 50M LOGGING EXTENT MANAGEMENT LOCAL
```

```
SEGMENT SPACE MANAGEMENT AUTO;
```

**Penjelasan command:** Berfungsi untuk membuat tablespace baru bernama tbsalert dengan data file berukuran 50MB.

**Penjelasan error:** Error terjadi karena tablespace "TBSALERT" sudah ada di database.

```

SQL> CREATE TABLESPACE tbsalert
DATAFILE '/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'
SIZE 50M LOGGING EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
2     3     4 CREATE TABLESPACE tbsalert
*
ERROR at line 1:
ORA-01543: tablespace 'TBSALERT' already exists
Help: https://docs.oracle.com/error-help/db/ora-01543/

```

**select \* from dba\_data\_files**

**where TABLESPACE\_NAME='TBSALERT'; --52428800**

**Penjelasan command:** Berfungsi untuk menampilkan informasi mengenai datafile yang terkait dengan tablespace "TBSALERT".

```

SQL> select * from dba_data_files
where TABLESPACE_NAME='TBSALERT';
2
FILE_NAME
-----
FILE_ID TABLESPACE_NAME          BYTES      BLOCKS STATUS
-----  

RELATIVE_FNO AUT    MAXBYTES  MAXBLOCKS INCREMENT_BY USER_BYTES USER_BLOCKS
-----  

ONLINE_LOST_WR
-----  

/opt/oracle/oradata/FREE/tbsalert.dbf
    16 TBSALERT            26214400      3200 AVAILABLE
    1024 NO              0           0           0   19890176      2428
ONLINE OFF

```

**select \* from dba\_free\_space**

**where TABLESPACE\_NAME='TBSALERT';--51380224**

**Penjelasan command:** Berfungsi untuk menampilkan informasi free space yang tersedia di dalam tablespace "TBSALERT".

```

SQL> select * from dba_free_space
where TABLESPACE_NAME='TBSALERT';
2
TABLESPACE_NAME          FILE_ID  BLOCK_ID      BYTES      BLOCKS
-----  

RELATIVE_FNO
-----  

TBSALERT                16        784      19791872      2416
    1024

```

**SELECT df.tablespace\_name tablespace, fs.bytes free, df.bytes,**

**fs.bytes\*100/ df.bytes PCT\_FREE**

**FROM dba\_data\_files df ,dba\_free\_space fs**

**WHERE df.tablespace\_name = fs.tablespace\_name**

**AND df.tablespace\_name = 'TBSALERT';**

**Penjelasan command:** Berfungsi untuk menghitung dan menampilkan persentase ruang kosong "PCT\_FREE" di tablespace "TBSALERT" dengan membandingkan total ruang kosong (dari dba\_free\_space) dengan total ukuran datafile dari dba\_data\_files.

```
SQL> SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
   fs.bytes*100/ df.bytes PCT_FREE
   FROM dba_data_files df ,dba_free_space fs
   WHERE df.tablespace_name = fs.tablespace_name
   AND df.tablespace_name = 'TBSALERT';
  2   3   4   5
TABLESPACE                      FREE        BYTES      PCT_FREE
-----  -----  -----
TBSALERT                         19791872    26214400       75.5
```

```
begin
DBMS_SERVER_ALERT.SET_THRESHOLD(
metrics_id => dbms_server_alert.tablespace_pct_full,
warning_operator =>DBMS_SERVER_ALERT.OPERATOR_GE,
warning_value => '55',
critical_operator =>DBMS_SERVER_ALERT.OPERATOR_GE,
critical_value => '70',
observation_period => 1,
consecutive_occurrences => 1,
instance_name => 'orcl',
object_type =>DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
object_name => 'TBSALERT');
end;
```

**Penjelasan command:** Blok PL/SQL menggunakan paket DBMS\_SERVER\_ALERT untuk mengatur ambang batas/threshold peringatan untuk tablespace "TBSALERT". Peringatan ("warning") akan muncul jika tablespace terisi 55% atau lebih, dan peringatan kritis ("critical") akan muncul jika terisi 70% atau lebih.

```
SQL> begin
DBMS_SERVER_ALERT.SET_THRESHOLD(
metrics_id => dbms_server_alert.tablespace_pct_full,
warning_operator =>DBMS_SERVER_ALERT.OPERATOR_GE,
warning_value => '55',
critical_operator =>DBMS_SERVER_ALERT.OPERATOR_GE,
critical_value => '70',
observation_period => 1,
consecutive_occurrences => 1,
instance_name => 'orcl',
object_type =>DBMS_SERVER_ALERT.OBJECT_TYPE_TABLESPACE,
object_name => 'TBSALERT');
end;
  2   3   4   5   6   7   8   9   10  11  12  13  14
 15  /
```

PL/SQL procedure successfully completed.

```
SELECT warning_value, critical_value  
FROM dba_thresholds  
WHERE object_name='TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa view dba\_thresholds dan memverifikasi bahwa ambang batas/threshold "warning" (55) dan "critical" (70) telah berhasil diatur untuk "TBSALERT".

```
SQL> SELECT warning_value, critical_value  
FROM dba_thresholds  
WHERE object_name='TBSALERT';  
  
 2   3  
WARNING_VALUE  
-----  
CRITICAL_VALUE  
-----  
55  
70
```

```
SELECT * FROM dba_outstanding_alerts  
WHERE object_name='TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa apakah ada peringatan yang sedang aktif atau beredar untuk tablespace "TBSALERT".

```
SQL> SELECT * FROM dba_outstanding_alerts  
WHERE object_name='TBSALERT';  
 2  
no rows selected
```

```
create table test100 ( emp_id number, name varchar2(100) )  
tablespace TBSALERT;
```

**Penjelasan command:** Berfungsi untuk membuat tabel baru "test100" dan secara spesifik menempatkannya di dalam tablespace "TBSALERT".

```
SQL> create table test100 ( emp_id number, name varchar2(100) )  
tablespace TBSALERT;  
 2  
Table created.
```

```
begin  
for i in 1..1000000  
loop  
insert into test100 values ( i, 'any dummy name' );  
end loop;  
commit;  
end;
```

**Penjelasan command:** Blok PL/SQL ini menjalankan looping untuk memasukkan 1.000.000 baris data ke dalam tabel test100.

**Penjelasan error:** Terjadi error karena tablespace TBSALERT kehabisan ruang (penuh) sebelum looping selesai.

```
SQL> begin
  for i in 1..1000000
  loop
    insert into test100 values ( i, 'any dummy name' );
  end loop;
  commit;
end;
2   3   4   5   6   7   8   /
begin
*
ERROR at line 1:
ORA-01653: unable to increase tablespace TBSALERT by 1MB during insert or
update on table SYS.TEST100
ORA-06512: at line 4
Help: https://docs.oracle.com/error-help/db/ora-01653/
```

```
SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
fs.bytes*100/ df.bytes PCT_FREE
FROM dba_data_files df ,dba_free_space fs
WHERE df.tablespace_name = fs.tablespace_name
AND df.tablespace_name = 'TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa kembali persentase ruang kosong di "TBSALERT" setelah upaya "insert" gagal karena tablespace penuh.

```
SQL> SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
  fs.bytes*100/ df.bytes PCT_FREE
  FROM dba_data_files df ,dba_free_space fs
 WHERE df.tablespace_name = fs.tablespace_name
 AND df.tablespace_name = 'TBSALERT';
 2   3   4   5
TABLESPACE                      FREE      BYTES      PCT_FREE
-----  -----  -----
TBSALERT                         917504    26214400        3.5
```

--Wait a few minutes. 10-15

```
SELECT reason, message_type , message_level
FROM dba_outstanding_alerts
WHERE object_name='TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa kembali apakah ada peringatan (alert) yang aktif, setelah tablespace "TBSALERT" terisi penuh.

```
SQL> SELECT reason, message_type , message_level
  FROM dba_outstanding_alerts
 WHERE object_name='TBSALERT';
 2   3
no rows selected
```

=====

```
show user;
```

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;
USER is "SYS"
SQL>
```

```
show con_name;
```

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;

CON_NAME
-----
FREEPDB1
```

```
drop TABLESPACE tbsalert INCLUDING CONTENTS and DATAFILES;
```

**Penjelasan command:** Berfungsi untuk menghapus tablespace "tbsalert" beserta seluruh isinya dan datafilesnya.

```
SQL> drop TABLESPACE tbsalert INCLUDING CONTENTS and DATAFILES;

Tablespace dropped.
```

```
CREATE TABLESPACE tbsalert
```

```
DATAFILE '/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'
```

```
SIZE 50M LOGGING EXTENT MANAGEMENT LOCAL
```

```
SEGMENT SPACE MANAGEMENT AUTO;
```

**Penjelasan command:** Berfungsi untuk membuat kembali tablespace "tbsalert" dengan ukuran 50MB.

```
SQL> CREATE TABLESPACE tbsalert
DATAFILE '/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'
SIZE 50M LOGGING EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
2 3 4
Tablespace created.
```

```
SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
fs.bytes*100/ df.bytes PCT_FREE
```

```
FROM dba_data_files df ,dba_free_space fs
```

```
WHERE df.tablespace_name = fs.tablespace_name
```

```
AND df.tablespace_name = 'TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa persentase ruang kosong di tablespace "TBSALERT" yang baru dibuat.

```

SQL> SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
fs.bytes*100/ df.bytes PCT_FREE
FROM dba_data_files df ,dba_free_space fs
WHERE df.tablespace_name = fs.tablespace_name
AND df.tablespace_name = 'TBSALERT';
2   3   4   5
TABLESPACE                      FREE      BYTES    PCT_FREE
-----  -----  -----  -----
TBSALERT                         46071808  52428800     87.875

```

**create table test100 ( emp\_id number, name varchar2(100) )**

**tablespace TBSALERT;**

**Penjelasan command:** Blok PL/SQL ini kembali mencoba memasukkan 1.000.000 baris data ke tabel "test100".

```

SQL> create table test100 ( emp_id number, name varchar2(100) )
tablespace TBSALERT;

2
Table created.

```

```

begin
for i in 1..1000000
loop
insert into test100 values ( i, 'any dummy name' );
end loop;
commit;
end;

```

**Penjelasan command:** Blok PL/SQL ini kembali mencoba memasukkan 1.000.000 baris data ke tabel "test100".

```

SQL> begin
  for i in 1..1000000
  loop
  insert into test100 values ( i, 'any dummy name' );
  end loop;
  commit;
end;
2   3   4   5   6   7   8
9  /
PL/SQL procedure successfully completed.

```

```

SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
fs.bytes*100/ df.bytes PCT_FREE
FROM dba_data_files df ,dba_free_space fs
WHERE df.tablespace_name = fs.tablespace_name
AND df.tablespace_name = 'TBSALERT';

```

**Penjelasan command:** Berfungsi untuk memeriksa persentase ruang kosong di "TBSALERT" setelah diisi data (namun sebelum "delete").

```
SQL> SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
   fs.bytes*100/ df.bytes PCT_FREE
   FROM dba_data_files df ,dba_free_space fs
 WHERE df.tablespace_name = fs.tablespace_name
 AND df.tablespace_name = 'TBSALERT';
 2   3   4   5
TABLESPACE          FREE      BYTES    PCT_FREE
-----
TBSALERT           983040  52428800     1.875
TBSALERT          15728640  52428800      30
```

**delete from test100;**

**Penjelasan command:** Berfungsi untuk menghapus semua 1.000.000 baris data dari tabel "test100".

```
SQL> delete from test100;
1000000 rows deleted.
```

**commit;**

**Penjelasan command:** berfungsi untuk menyimpan semua perubahan yang telah dibuat dalam database secara permanen.

```
SQL> commit;
Commit complete.
```

```
SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
   fs.bytes*100/ df.bytes PCT_FREE
   FROM dba_data_files df ,dba_free_space fs
 WHERE df.tablespace_name = fs.tablespace_name
 AND df.tablespace_name = 'TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa persentase ruang kosong di "TBSALERT" setelah data dihapus, ruang kosong tidak kembali karena "delete" tidak melepaskan High Water Mark (HWM).

```

SQL> SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
fs.bytes*100/ df.bytes PCT_FREE
FROM dba_data_files df ,dba_free_space fs
WHERE df.tablespace_name = fs.tablespace_name
AND df.tablespace_name = 'TBSALERT';
 2   3   4   5
TABLESPACE                      FREE      BYTES      PCT_FREE
-----
TBSALERT                         983040    52428800     1.875
TBSALERT                         15728640   52428800     30

```

--Create a Segment Advisor Task

```

DECLARE
tname VARCHAR2(128) := 'my_seg_task3';
tname_desc VARCHAR2(128) := 'Get shrink advice for segments in TBSALERT';
task_id NUMBER;
object_id NUMBER;
objectname VARCHAR2(100);
objecttype VARCHAR2(100);
BEGIN
dbms_advisor.create_task('Segment Advisor',
task_id,tname,tname_desc,NULL);---select * from DBA_ADVISOR_DEFINITIONS
dbms_advisor.create_object(tname,'TABLESPACE','TBSALERT','','',NULL, ''
,object_id) ;
dbms_advisor.set_task_parameter(tname,'RECOMMEND_ALL','TRUE');
END;
```

**Penjelasan command:** Blok PL/SQL ini menggunakan paket dbms\_advisor untuk membuat task "Segment Advisor" bernama "my\_seg\_task3". Tugas ini diatur untuk menganalisis objek di dalam tablespace "TBSALERT" dan merekomendasikan semua perbaikan ("RECOMMEND\_ALL", "TRUE").

```

SQL> DECLARE
tname VARCHAR2(128) := 'my_seg_task3';
tname_desc VARCHAR2(128) := 'Get shrink advice for segments in TBSALERT';
task_id NUMBER;
object_id NUMBER;
objectname VARCHAR2(100);
objecttype VARCHAR2(100);
BEGIN
dbms_advisor.create_task('Segment Advisor', task_id,tname,tname_desc,NULL);---s
elect * from DBA_ADVISOR_DEFINITIONS
dbms_advisor.create_object(tname,'TABLESPACE','TBSALERT','','',NULL, '',obj
ect_id) ;
dbms_advisor.set_task_parameter(tname,'RECOMMEND_ALL','TRUE');
END;
 2   3   4   5   6   7   8   9   10   11   12   13
14  /
```

PL/SQL procedure successfully completed.

--Execute the task.

```
DECLARE
  tname VARCHAR2(128) := 'my_seg_task3';
BEGIN
  dbms_advisor.EXECUTE_TASK(tname);
END;
```

**Penjelasan command:** Blok PL/SQL ini mengeksekusi tugas "Segment Advisor" ("my\_seg\_task3") yang telah dibuat sebelumnya.

```
SQL> DECLARE
  tname VARCHAR2(128) := 'my_seg_task3';
BEGIN
  dbms_advisor.EXECUTE_TASK(tname);
END;
2   3   4   5   6  /
PL/SQL procedure successfully completed.

SQL> █
```

```
SELECT DESCRIPTION FROM dba_advisor_tasks WHERE
TASK_NAME='my_seg_task3';
```

**Penjelasan command:** Berfungsi untuk menampilkan deskripsi dari tugas "my\_seg\_task3" untuk memverifikasi pembuatannya.

```
SQL> SELECT DESCRIPTION FROM dba_advisor_tasks WHERE TASK_NAME='my_seg_task3';

DESCRIPTION
-----
Get shrink advice for segments in TBSALERT
```

```
SELECT attr1, attr2, message
FROM dba_advisor_findings f, dba_advisor_objects o
WHERE f.task_name = o.task_name
AND f.object_id = o.object_id
AND f.task_name = 'my_seg_task3';
```

**Penjelasan command:** Berfungsi untuk menampilkan temuan (findings) atau rekomendasi dari "Segment Advisor" (task "my\_seg\_task3"), dengan menggabungkan view dba\_advisor\_findings dan dba\_advisor\_objects.

```
SQL> SELECT attr1, attr2, message
  FROM dba_advisor_findings f, dba_advisor_objects o
  WHERE f.task_name = o.task_name
  AND f.object_id = o.object_id
  AND f.task_name = 'my_seg_task3';
2   3   4   5
ATTR1
-----
ATTR2
-----
MESSAGE
-----
SYS
TEST100
Enable row movement of the table SYS.TEST100 and perform shrink, estimated savings is 29351936 bytes.
```

```
alter table test100 enable row movement;
```

**Penjelasan command:** Berfungsi untuk mengizinkan pergerakan baris (row movement) pada tabel "test100". Ini adalah syarat wajib untuk melakukan operasi "shrink".

```
SQL>
alter table test100 enable row movement;

SQL>
Table altered.
```

```
ALTER TABLE test100 SHRINK SPACE;
```

**Penjelasan command:** Berfungsi untuk melakukan operasi "shrink" (penyusutan) pada tabel "test100". Perintah ini akan memadatkan data dan menurunkan High Water Mark (HWM) untuk melepaskan ruang kosong yang tidak terpakai.

```
SQL> ALTER TABLE test100 SHRINK SPACE;

Table altered.
```

```
SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
fs.bytes*100/ df.bytes PCT_FREE
FROM dba_data_files df ,dba_free_space fs
WHERE df.tablespace_name = fs.tablespace_name
AND df.tablespace_name = 'TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa kembali persentase ruang kosong di "TBSALERT" setelah operasi "shrink" berhasil melepaskan ruang.

```
SQL> SELECT df.tablespace_name tablespace, fs.bytes free, df.bytes,
fs.bytes*100/ df.bytes PCT_FREE
FROM dba_data_files df ,dba_free_space fs
WHERE df.tablespace_name = fs.tablespace_name
AND df.tablespace_name = 'TBSALERT';

      2      3      4      5
TABLESPACE                         FREE      BYTES      PCT_FREE
-----+-----+-----+-----+
TBSALERT                           46006272    52428800      87.75
-----+-----+-----+-----+
```

/\*  
Managing Resumable Space Allocation

A resumable statement:

- 1- Enables you to suspend large operations instead of receiving an error
- 2- Gives you a chance to fix the problem while the operation is suspended, rather than starting over

3-Is suspended for the following conditions:

>>Out of space  
>>Maximum extents reached  
>>Space quota exceeded

4-Can be suspended and resumed multiple times

\*/

**show user;**

**Penjelasan command:** Berfungsi untuk menampilkan user yang sedang aktif di sesi ini.

```
SQL> show user;
USER is "SYS"
SQL> █
```

**show con\_name;**

**Penjelasan command:** berfungsi untuk menampilkan koneksi database atau nama container yang sedang aktif.

```
SQL> show con_name;

CON_NAME
-----
FREEPDB1
```

**drop TABLESPACE tbsalert INCLUDING CONTENTS and DATAFILES;**

**Penjelasan command:** Berfungsi untuk menghapus tablespace "tbsalert" beserta seluruh isinya dan datafilesnya.

```
SQL> drop TABLESPACE tbsalert INCLUDING CONTENTS and DATAFILES;

Tablespace dropped.
```

**CREATE TABLESPACE tbsalert**

**DATAFILE '/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'**  
**SIZE 10M LOGGING EXTENT MANAGEMENT LOCAL**  
**SEGMENT SPACE MANAGEMENT AUTO;**

**Penjelasan command:** Berfungsi untuk membuat kembali tablespace "tbsalert" dengan ukuran yang jauh lebih kecil (10MB).

```
SQL> CREATE TABLESPACE tbsalert
DATAFILE '/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'
SIZE 10M LOGGING EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;

2    3    4
Tablespace created.
```

```
create table test100 ( emp_id number, name varchar2(100) )
tablespace TBSALERT;
```

**Penjelasan command:** Berfungsi untuk membuat tabel "test100" di dalam tablespace "TBSALERT" yang berukuran 10MB.

```
SQL> create table test100 ( emp_id number, name varchar2(100) )
  tablespace TBSALERT;
  2
Table created.
```

```
begin
for i in 1..400000
loop
insert into test100 values ( i, 'any dummy name' );
commit;
end loop;
end;
```

**Penjelasan command:** Blok PL/SQL ini mencoba memasukkan 400.000 baris data ke tabel "test100", dengan "commit" di setiap iterasi.

**Penjelasan error:** Error terjadi karena karena tablespace (10MB) akan penuh.

```
SQL> begin
  for i in 1..400000
  loop
    insert into test100 values ( i, 'any dummy name' );
    commit;
  end loop;
end;
  2   3   4   5   6   7   8   /
begin
*
ERROR at line 1:
ORA-01653: unable to increase tablespace TBSALERT by 1MB during insert or
update on table SYS.TEST100
ORA-06512: at line 4
Help: https://docs.oracle.com/error-help/db/ora-01653/
```

```
select * from test100;
```

**Penjelasan command:** Berfungsi untuk menghitung dan menampilkan jumlah baris yang berhasil dimasukkan ke "test100" sebelum error kehabisan ruang terjadi.

```
107701 rows selected.
```

```
/*
```

A statement executes in resumable mode only if its session has been enabled by one of the following actions:

- 1- The RESUMABLE\_TIMEOUT initialization parameter is set to a nonzero value.
- 2- An ALTER SESSION ENABLE RESUMABLE statement is issued

```
*/
```

```
ALTER SESSION ENABLE RESUMABLE;
```

**Penjelasan command:** Berfungsi untuk mengaktifkan mode "Resumable Space Allocation" untuk sesi ini. Jika terjadi error kehabisan ruang, operasi akan ditangguhkan (suspend) alih-alih gagal total.

```
SQL> ALTER SESSION ENABLE RESUMABLE;  
Session altered.
```

```
truncate table test100;
```

**Penjelasan command:** Berfungsi untuk menghapus semua data di tabel "test100" dengan cepat (menurunkan HWM), untuk persiapan pengujian mode "resumable".

```
SQL> truncate table test100;  
Table truncated.
```

```
begin  
for i in 1..400000  
loop  
insert into test100 values ( i, 'any dummy name' );  
commit;  
end loop;  
end;
```

**Penjelasan command:** Blok PL/SQL ini kembali mencoba memasukkan 400.000 baris data. Karena mode "resumable" aktif, sesi ini akan ditangguhkan (suspend) ketika tablespace "TBSALERT" penuh, menunggu admin memperbaiki masalah ruang.

```
SQL> begin  
for i in 1..400000  
loop  
insert into test100 values ( i, 'any dummy name' );  
commit;  
end loop;  
end;  
2   3   4   5   6   7   8  /  
  
/  
  
PL/SQL procedure successfully completed.
```

--the script will be suspend 7,200 seconds=2 hours. ( this the default )  
--you can do ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600  
--now oracle will not give us any message that this script has been suspend  
--but we can see it as DBA. open new session and do this  
/\*

```
select * from dba_resumable;
```

**Penjelasan command:** Berfungsi untuk menampilkan informasi tentang sesi-sesi yang sedang ditangguhkan (suspended) karena menunggu alokasi ruang.

```
SQL> select * from dba_resumable;
  2  on table SYS.TEST100
  3
  4    USER_ID SESSION_ID INSTANCE_ID COORD_INSTANCE_ID COORD_SESSION_ID STATUS
  5    -----
  6    TIMEOUT START_TIME           SUSPEND_TIME          RESUME_TIME
  7
  8    NAME
  9
 10   SQL_TEXT
 11
 12   ERROR_NUMBER
 13
 14
 15   ERROR_NUMBER
 16
 17   ERROR_PARAMETER1
 18
 19   ERROR_PARAMETER2
 20
 21   ERROR_PARAMETER3
 22
 23   ERROR_PARAMETER4
 24   ERROR_MSG
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
```

```
SELECT file_name, autoextensible FROM dba_data_files WHERE
tablespace_name='TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memeriksa status file data "TBSALERT", khususnya apakah fitur "autoextensible" (perluasan otomatis) aktif atau tidak ("NO").

```
SQL> SELECT file_name, autoextensible FROM dba_data_files WHERE
tablespace_name='TBSALERT';
 2
FILE_NAME
---
AUT
---
/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf
NO
```

**ALTER DATABASE DATAFILE**

```
'/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'
```

**AUTOEXTEND ON MAXSIZE unlimited;**

**Penjelasan command:** Berfungsi untuk memperbaiki masalah ruang dengan mengaktifkan "AUTOEXTEND ON" (perluasan otomatis) pada file data "TBSALERT". Setelah ini dieksekusi, sesi yang ditangguhkan akan otomatis melanjutkan (resume) operasinya.

```
SQL> ALTER DATABASE DATAFILE
  '/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf'
AUTOEXTEND ON MAXSIZE unlimited;
  2   3
Database altered.
```

```
SELECT file_name, autoextensible FROM dba_data_files WHERE
tablespace_name='TBSALERT';
```

**Penjelasan command:** Berfungsi untuk memverifikasi bahwa file data "TBSALERT" sekarang sudah "autoextensible" ("YES").

```
SQL> SELECT file_name, autoextensible FROM dba_data_files WHERE
tablespace_name='TBSALERT';
  2
FILE_NAME
-----
AUT
-----
/opt/oracle/oradata/FREE/pdbts/tbsalert.dbf
YES
```

--WAIT until NO ROWS retrieved by THIS QUERY

```
select * from dba_resumable;
```

**Penjelasan command:** Berfungsi untuk memeriksa kembali view dba\_resumable. Setelah operasi "insert" berhasil dilanjutkan dan selesai, tidak akan ada baris lagi di view ini.

```
-----  
USER_ID SESSION_ID INSTANCE_ID COORD_INSTANCE_ID COORD_SESSION_ID STATUS  
TIMEOUT START_TIME SUSPEND_TIME RESUME_TIME  
-----  
NAME  
-----  
SQL_TEXT  
ERROR_NUMBER  
-----  
ERROR_PARAMETER1  
ERROR_PARAMETER2  
-----  
ERROR_PARAMETER3  
-----  
ERROR_PARAMETER4  
-----  
ERROR_PARAMETERS5  
-----  
ERROR_MSG  
-----  
SQL> ■
```

```
select count(1) from test100;
```

**Penjelasan command:** Berfungsi untuk menghitung total baris di "test100" untuk memverifikasi bahwa seluruh 400.000 baris berhasil dimasukkan setelah sesi dilanjutkan.

```
SQL> select count(1) from test100;  
  
 COUNT(1)  
-----  
 800000
```

```
*/
```