TABLESPACE MANAGMENT

Tablespaces are the storage units in an oracle database to store the data. Each tablespace in an oracle database consists of one or more files called datafiles. Datafiles are the actual data records, and the information stored in datafiles can be accessed or modified later whenever necessary.

Resizing a tablespace in an Oracle RAC (Real Application Clusters) database involves increasing or decreasing the size of the datafiles associated with that tablespace. Here’s a step-by-step guide to perform this operation:

### 1. Connect to the Database

 First, connect to your Oracle RAC database as a user with the necessary privileges (usually `SYSDBA`)

* sqlplus / as sysdba

### 2. Check the Current Size of the Tablespace

You can query the current size and other details of the tablespace you intend to resize.

* SELECT TABLESPACE\_NAME, FILE\_NAME, BYTES/1024/1024 AS SIZE\_MB

FROM DBA\_DATA\_FILES

WHERE TABLESPACE\_NAME = 'YOUR\_TABLESPACE\_NAME';

### 3. Resize the Datafile

To resize a datafile, you use the `ALTER DATABASE` command. Here’s how you can increase or decrease the size of a datafile:

#### Increasing the Size of a Datafile

You can increase the size of a datafile using the `RESIZE` option. For example, to increase a datafile to 2GB:

* ALTER DATABASE DATAFILE '/path/to/your/datafile.dbf' RESIZE 2G;

#### Adding a New Datafile

Alternatively, you can add a new datafile to the tablespace if you prefer not to resize the existing ones:

* ALTER TABLESPACE YOUR\_TABLESPACE\_NAME
* ADD DATAFILE '/path/to/your/newdatafile.dbf' SIZE 1G;

### 4. Decreasing the Size of a Datafile

If you need to reduce the size of a datafile, ensure the space to be released is not currently used by the tablespace. This usually involves moving data around, which can be complex and risky. Here is an example of how you can decrease the size:

1. Identify the high watermark:

* SELECT FILE\_NAME, BLOCK\_ID, BLOCKS

FROM DBA\_EXTENTS

WHERE TABLESPACE\_NAME = 'YOUR\_TABLESPACE\_NAME'

ORDER BY FILE\_NAME, BLOCK\_ID DESC;

2. Reduce the size (ensure the new size is larger than the high watermark):

* ALTER DATABASE DATAFILE '/path/to/your/datafile.dbf' RESIZE new\_size;

### 5. Verify the Changes

After resizing, verify the new size of the tablespace:

* SELECT TABLESPACE\_NAME, FILE\_NAME, BYTES/1024/1024 AS SIZE\_MB

FROM DBA\_DATA\_FILES

WHERE TABLESPACE\_NAME = 'YOUR\_TABLESPACE\_NAME';

### Additional Considerations

- \*\*Monitoring Free Space\*\*: Use the `DBA\_FREE\_SPACE` view to monitor free space in the tablespace:

* SELECT TABLESPACE\_NAME, FILE\_ID, BYTES/1024/1024 AS FREE\_MB

    FROM DBA\_FREE\_SPACE ;

- \*\*Autoextend\*\*: If the datafile is set to autoextend, it will automatically grow when it reaches its limit. You can check and modify this setting:

* ALTER DATABASE DATAFILE '/path/to/your/datafile.dbf' AUTOEXTEND ON NEXT 100M MAXSIZE UNLIMITED;

- \*\*RAC Considerations\*\*: Ensure all RAC instances are aware of the changes. Typically, changes are propagated automatically, but verify and monitor the database to ensure consistency across instances.

**How to monitor Tablespace Growth?**

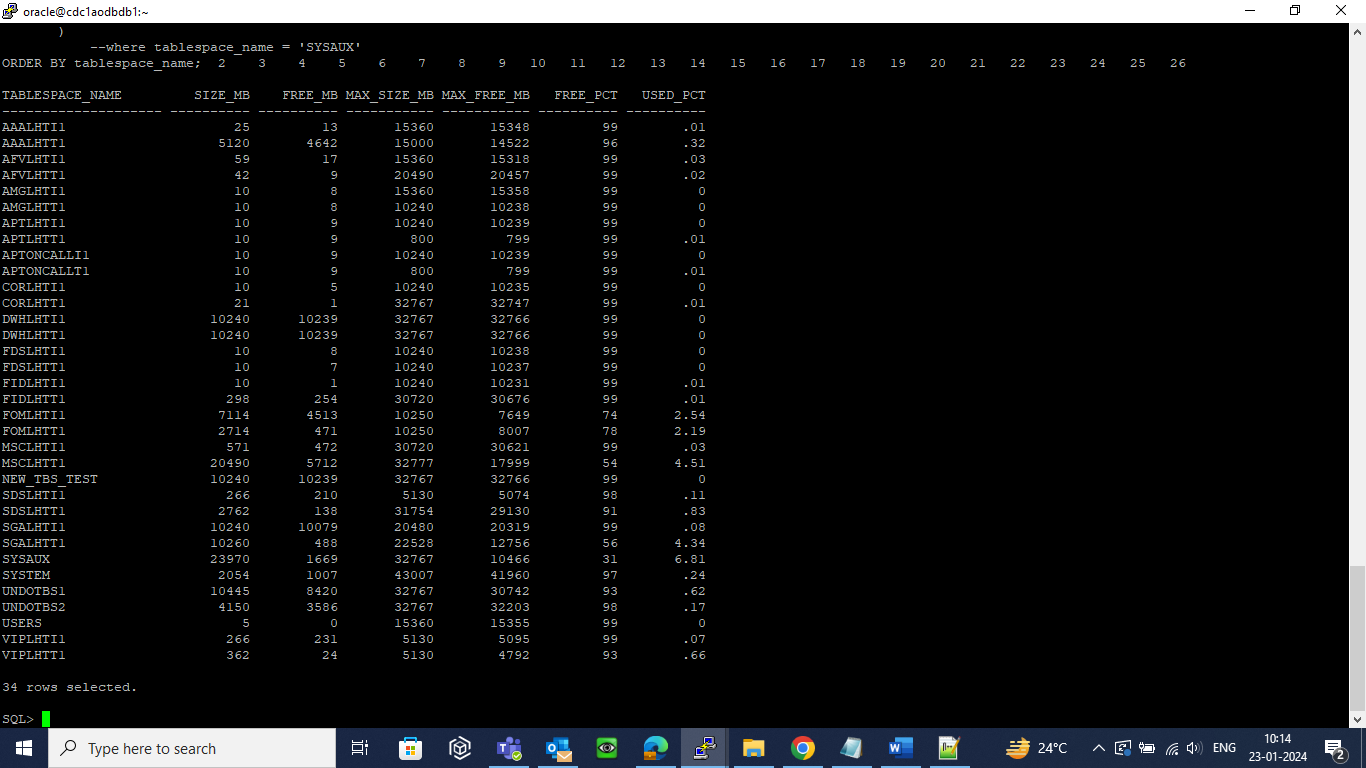
To monitor the tablespace growth, you need to analyze the tablespace tables or views using the script

below:

**Step 1: Analyzing TABLESPACE Views:**

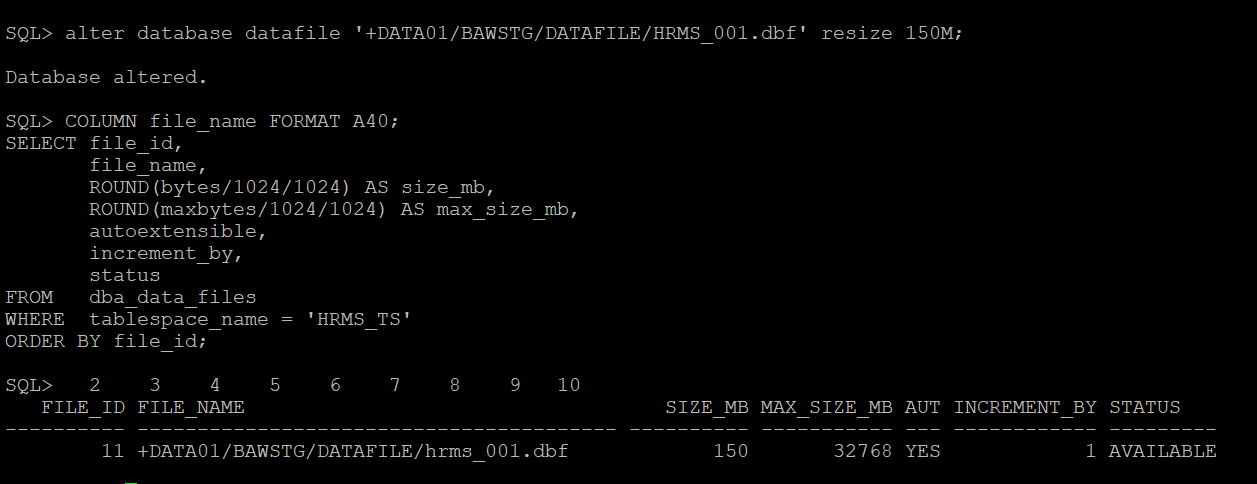
* Run the following script to analyze the tablespace’s storage information:  
    
  SET PAGESIZE 140 LINESIZE 200;  
  SELECT tablespace\_name,  
         size\_mb,  
         free\_mb,  
         max\_size\_mb,  
         max\_free\_mb,  
         TRUNC((max\_free\_mb/max\_size\_mb) \* 100) AS free\_pct,  
         ROUND((max\_size\_mb-max\_free\_mb)/max\_size\_mb\*10,2) AS used\_pct  
  FROM   (  
          SELECT a.tablespace\_name,  
                 b.size\_mb,  
                 a.free\_mb,  
                 b.max\_size\_mb,  
                 a.free\_mb + (b.max\_size\_mb - b.size\_mb) AS max\_free\_mb  
          FROM   (SELECT tablespace\_name,  
                         TRUNC(SUM(bytes)/1024/1024) AS free\_mb  
                  FROM dba\_free\_space  
                  GROUP BY tablespace\_name) a,  
                 (SELECT tablespace\_name,  
                         TRUNC(SUM(bytes)/1024/1024) AS size\_mb,  
                         TRUNC(SUM(GREATEST(bytes,maxbytes))/1024/1024) AS max\_size\_mb  
                  FROM dba\_data\_files  
                  GROUP BY tablespace\_name) b  
          WHERE  a.tablespace\_name = b.tablespace\_name  
         )   
       --where tablespace\_name = 'SYSAUX'  
  ORDER BY tablespace\_name;

* The **TABLESPACE** script output is displayed, as shown below.

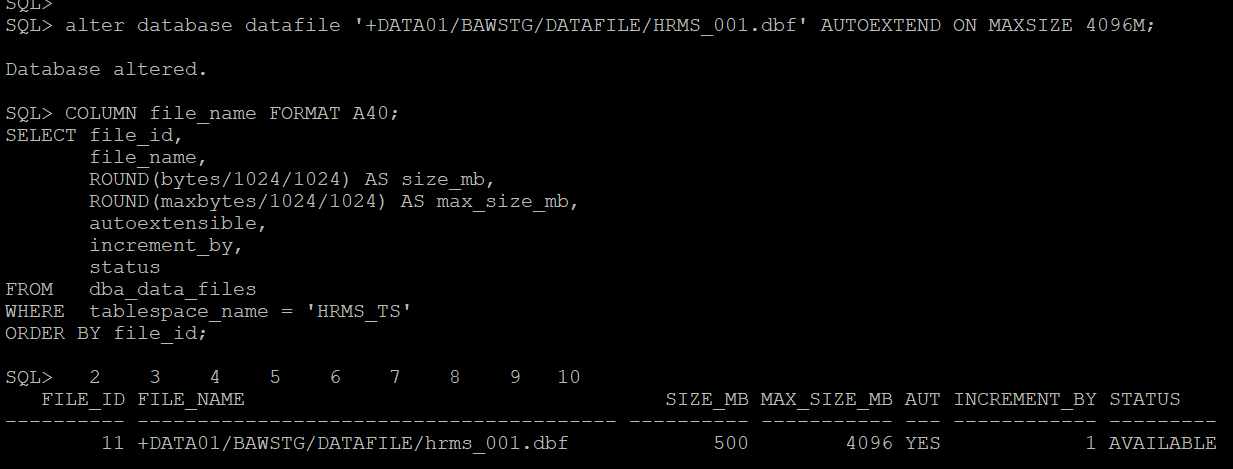


The **TABLESPACE** script output

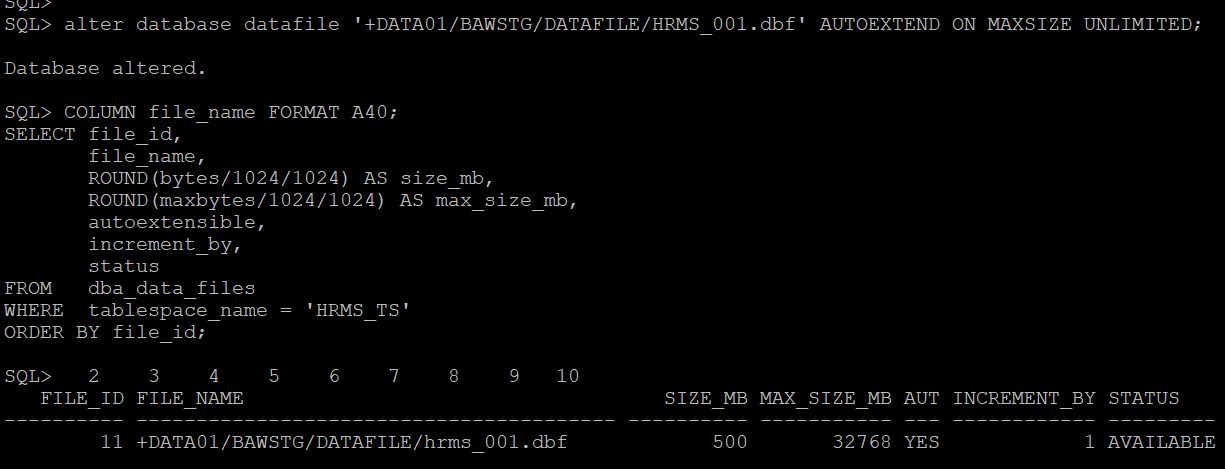
* The script output contains the following:   
  a. Tablespace Name  
  b. Tablespace Size  
  c. Available Free Space Size  
  f. Maximum Extensible Size  
  g. Available Free Space Percentage   
  h. Used Space Percentage
* Resize the tablespace if it is approaching the maximum threshold value. The below command is used to resize the tablespace.  
    
  *SQL>* ALTER DATABASE DATAFILE ‘ C:\ORACLE\ORDATA\IDMAKER\DMKR\_ASLINE.DBF’ RESIZE 150M;



* Resize datafile if the current datafile size is below the target max size. The below commands are used to resize the datafile.  
    
  **Increasing the size limit**  
  *SQL>*

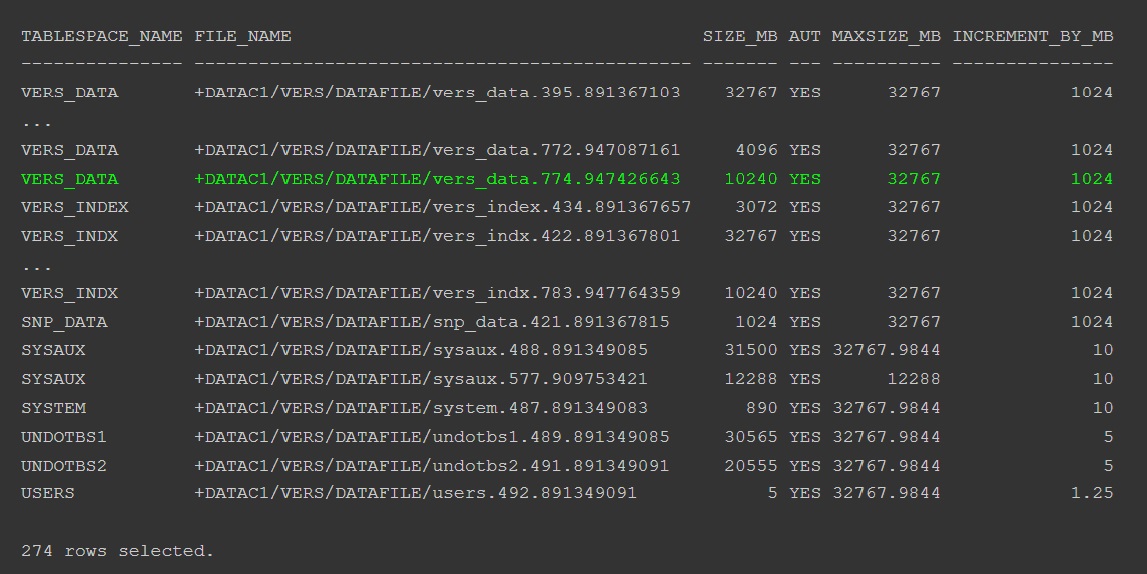
ALTER DATABASE DATAFILE ‘C:\ORACLE\ORDATA\IDMAKER\DMKR\_ASLINE.DBF’ AUTOEXTEND ON MAXSIZE 4096M;  
   


* **Removing the size limit**  
  *SQL>*

ALTER DATABASE DATAFILE ‘C:\ORACLE\ORDATA\IDMAKER\DMKR\_ASLINE.DBF’ AUTOEXTEND ON MAXSIZE UNLIMITED;   
 

* Add new datafile if the current datafile size is reached target max storage. Use the below command to add a datafile.   
    
  *SQL>*

ALTER TABLESPACE VERS\_DATA ADD DATAFILE '+DATAC1' SIZE 10G AUTOEXTEND ON NEXT 1G MAXSIZE 32767M;



Adding a new datafile