

select segment_name, sum(bytes)/1024/1024 from USER_segments group by segment_name;
Oracle Data Dictionary Views

Dynamic Performance Views

These views are called dynamic performance views because they are continuously updated while a database is open and in use. The views, also sometimes called V\$ views, contain

System and session parameters

Memory usage and allocation

File states (including RMAN backup files)

Oracle Data Dictionary Views

Dynamic Performance Views

Progress of jobs and tasks

SQL execution

Statistics and metrics

e.g., V\$SESSION, V\$DATAFILE etc.

End

Introduction to Database Administration

TOPIC 49

Memory Management & Memory Architecture Overview

Memory Management & Memory Architecture Overview

Memory & Architecture

Oracle Memory Management

Automatic Memory Management

Adjust Memory Target Manually

Enable Automatic Shared Memory

Memory Management & Memory Architecture Overview

Oracle Memory Management

Memory management involves maintaining optimal sizes for the Oracle Database instance memory structures as demands on the database change.

The memory structures that must be managed are the system global area (SGA) and the instance program global area (instance PGA).

Memory Management & Memory Architecture Overview

Oracle Memory Management

Oracle recommends that you enable the method known as automatic memory management.

Memory Management & Memory Architecture Overview

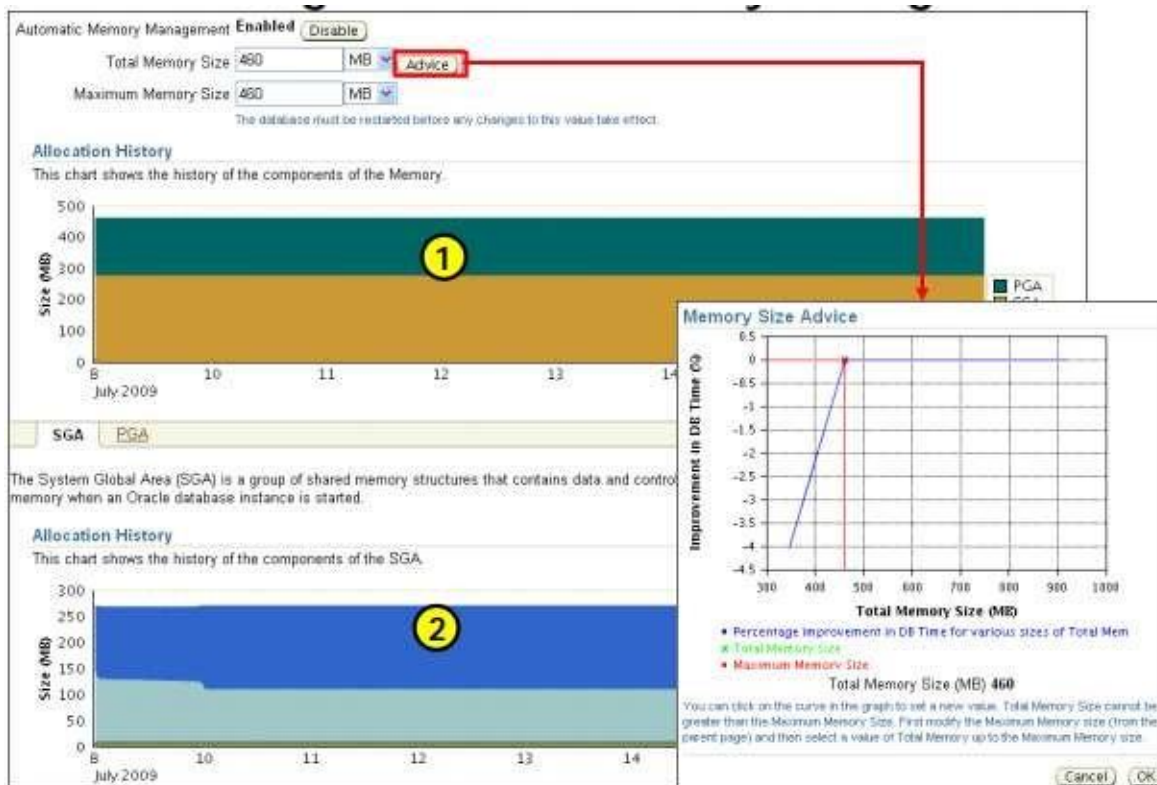
Automatic Memory Management

You designate only the total memory size to be used by the instance, and Oracle Database dynamically exchanges memory between the SGA and the instance PGA as needed to meet processing demands.

If you prefer to exercise more direct control over the sizes of individual memory components, you can disable automatic memory management and configure the database for manual memory management

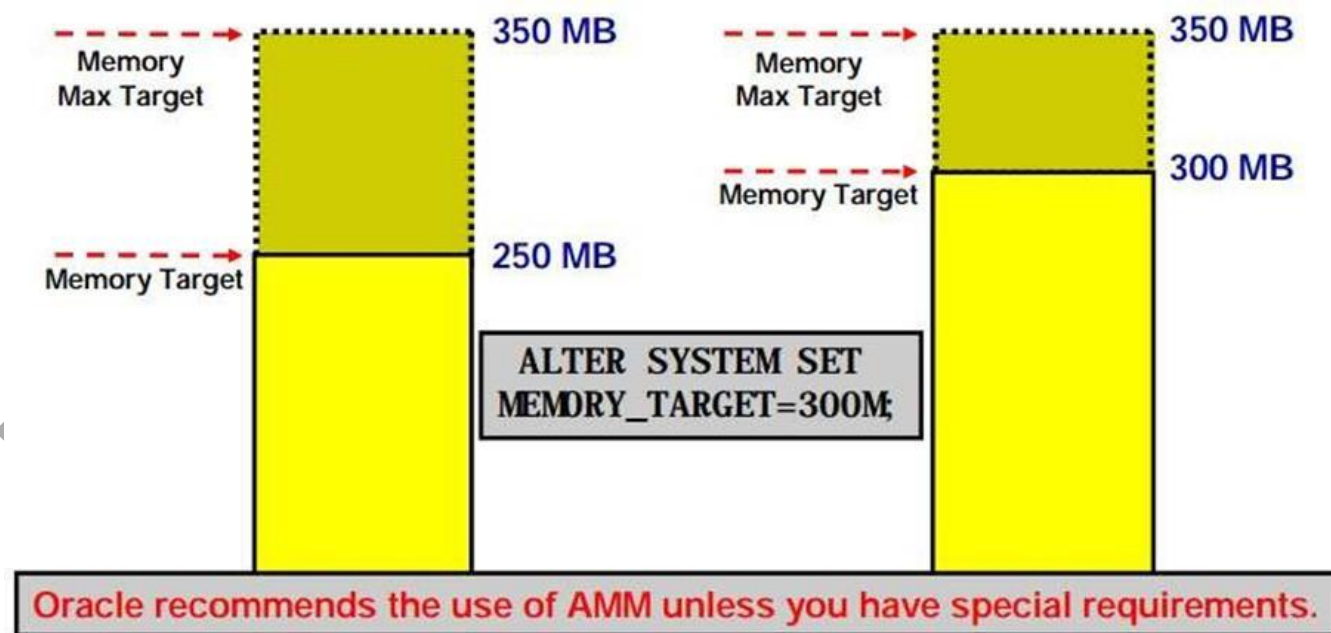
Memory Management & Memory Architecture Overview

Automatic Memory Management



Memory Management & Memory Architecture Overview

Adjust Memory Target Manually



Memory Management & Memory Architecture Overview

Enable Automatic Shared Memory

To enable ASMM from manual shared memory management:

1. Get a value for **SGA_TARGET**:

```
SELECT ((SELECT SUM(value) FROM V$SGA) - (SELECT CURRENT_SIZE  
FROM V$SGA_DYNAMIC_FREE_MEMORY)) "SGA_TARGET" FROM DUAL;
```

2. Use that value to set **SGA_TARGET**.
3. Set the values of the automatically sized SGA components to 0.

To switch to ASMM from Automatic Memory Management:

1. Set the **MEMORY_TARGET** initialization parameter to 0.
2. Set the values of the automatically sized SGA components to 0.

End

Introduction to Database Administration

TOPIC 50

Memory Management & Memory Architecture Overview

Memory Management & Memory Architecture Overview

Memory Management & Architecture

Memory Architecture Overview

Oracle Database Memory Structure

Current Sizes of SGA_TARGET and SGA_AGGREGATE_TARGET

V\$MEMORY_TARGET_ADVICE provides tuning advice for the MEMORY_TARGET

V\$PGASTAT

Memory Management & Memory Architecture Overview

Memory Architecture Overview

System Global Area (SGA)

The SGA is shared by all server and background processes. Examples of data stored in the SGA include cached data blocks and shared SQL areas.

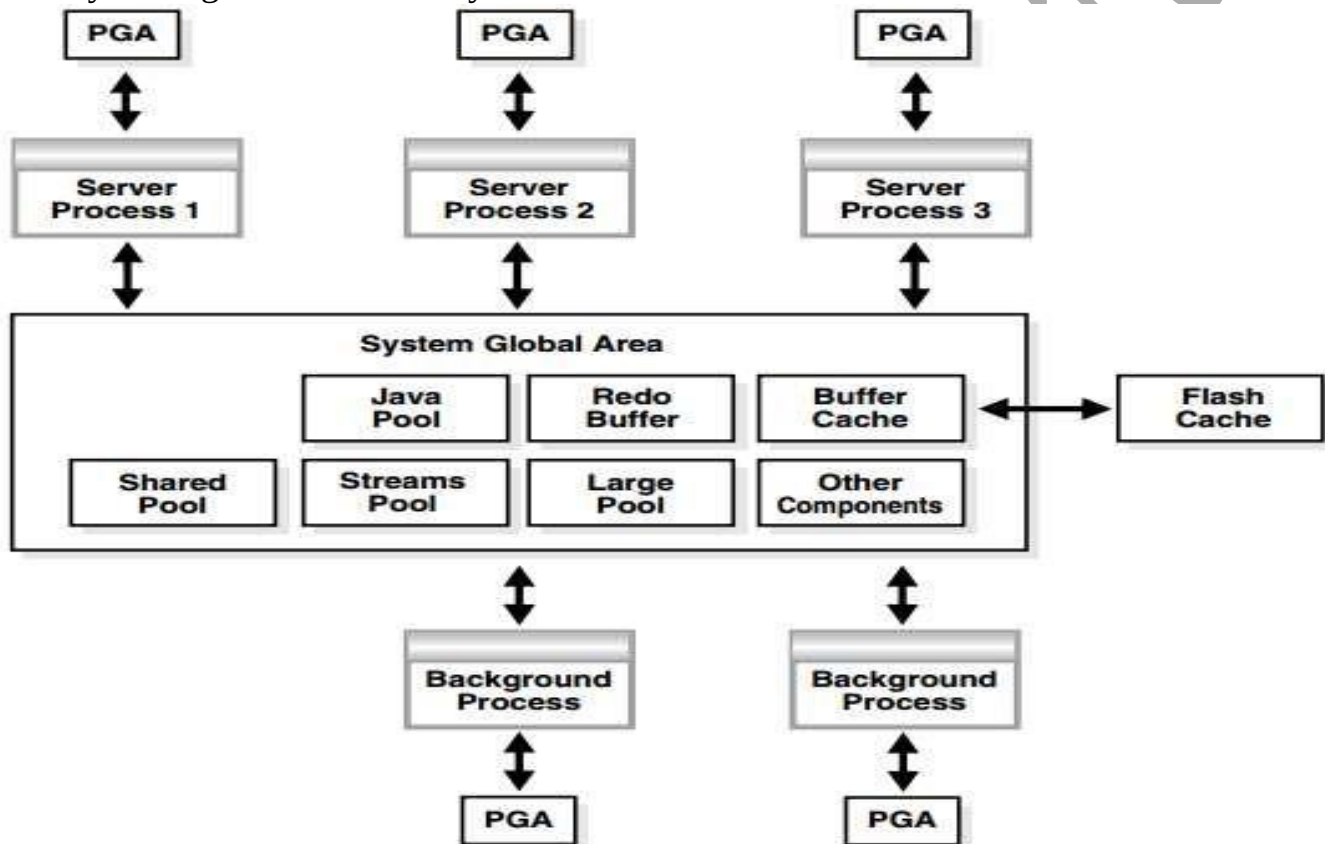
Memory Management & Memory Architecture Overview

Memory Architecture Overview

Program Global Area (PGA)

A PGA is a memory region that contains data and control information for a server process. It is non shared memory created by Oracle Database when a server process is started. Access to the PGA is exclusive to the server process. There is one PGA for each server process.

Memory Management & Memory Architecture Overview



Oracle Database Memory Structure

Memory Management & Memory Architecture Overview

Current Sizes of SGA_TARGET and SGA_AGGREGATE_TARGET

NAME	TYPE	VALUE
archive_lag_target	integer	0
db_flashback_retention_target	integer	1440
fast_start_io_target	integer	0
fast_start_mttr_target	integer	0
memory_max_target	big integer	0
memory_target	big integer	0
parallel_servers_target	integer	16
pga_aggregate_target	big integer	90M
sga_target	big integer	272M

Run the following query to determine the maximum instance PGA allocated since the database was started:

```
select value from v$pgastat where name='maximum PGA allocated';
```

Compute the maximum value between the query result from step 2b and PGA_AGGREGATE_TARGET. Add SGA_TARGET to this value.

```
memory_target = sga_target + max(pga_aggregate_target, maximum PGA allocated)
```

SHOW PARAMETER TARGET

Memory Management & Memory Architecture Overview

V\$MEMORY_TARGET_ADVICE

provides tuning advice for the MEMORY_TARGET

```
SQL> select * from v$memory_target_advice order by memory_size;
```

MEMORY_SIZE	MEMORY_SIZE_FACTOR	ESTD_DB_TIME	ESTD_DB_TIME_FACTOR	VERSION
180	.5	458	1.344	0
270	.75	367	1.0761	0
360	1	341	1	0
450	1.25	335	.9817	0
540	1.5	335	.9817	0
630	1.75	335	.9817	0

Memory Management & Memory Architecture Overview

V\$PGASTAT

```
select value from v$pgastat
```

where name='maximum PGA allocated'; For example, if SGA_TARGET is 272M and PGA_AGGREGATE_TARGET is 90M as

shown above, and

if the maximum PGA allocated is determined to be 120M, then MEMORY_TARGET should be at least 392M (272M + 120M).

Introduction to Database Administration

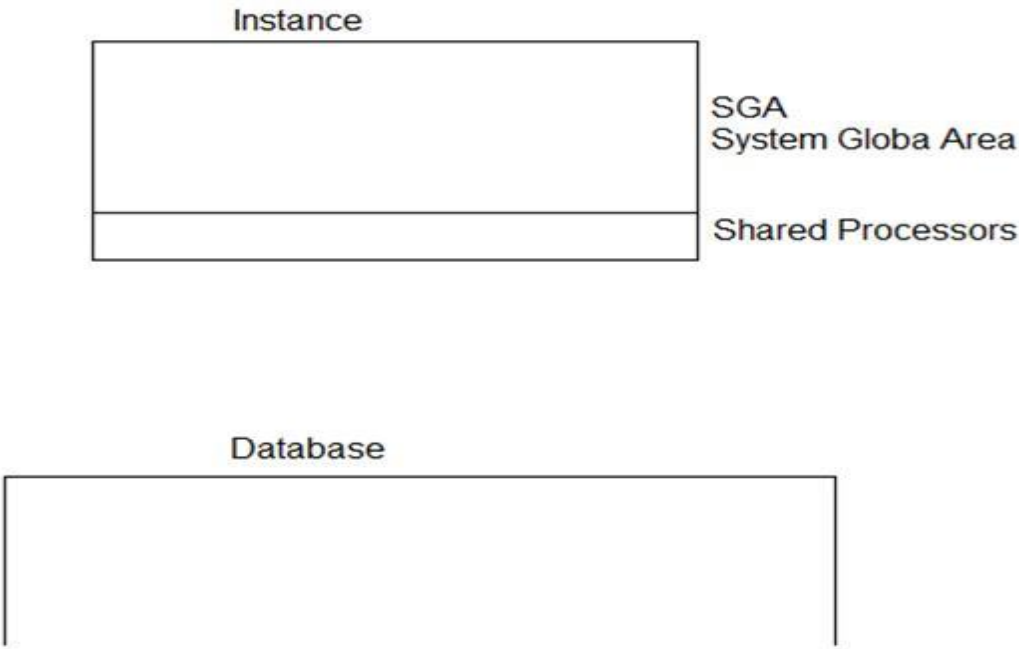
TOPIC 51

Memory Architecture Overview (SGA)

Memory Architecture Overview

Memory Management & Architecture

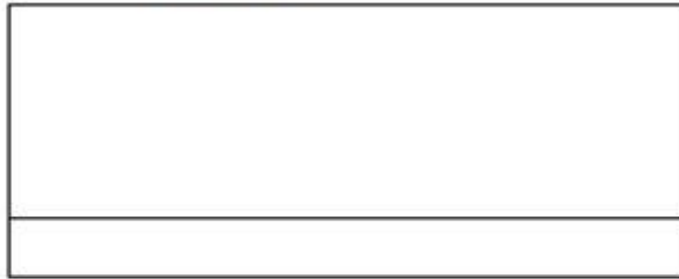
- Instance & Database
- Overview of SGA
- Overview of SGA & Database
- Memory Architecture Overview
- Instance & Database



Memory Architecture Overview

Overview of SGA

Instance

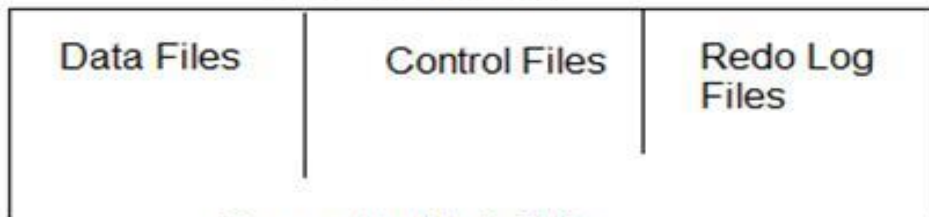


SGA
System Global Area

Shared Processors



Database (Files)



Data Files

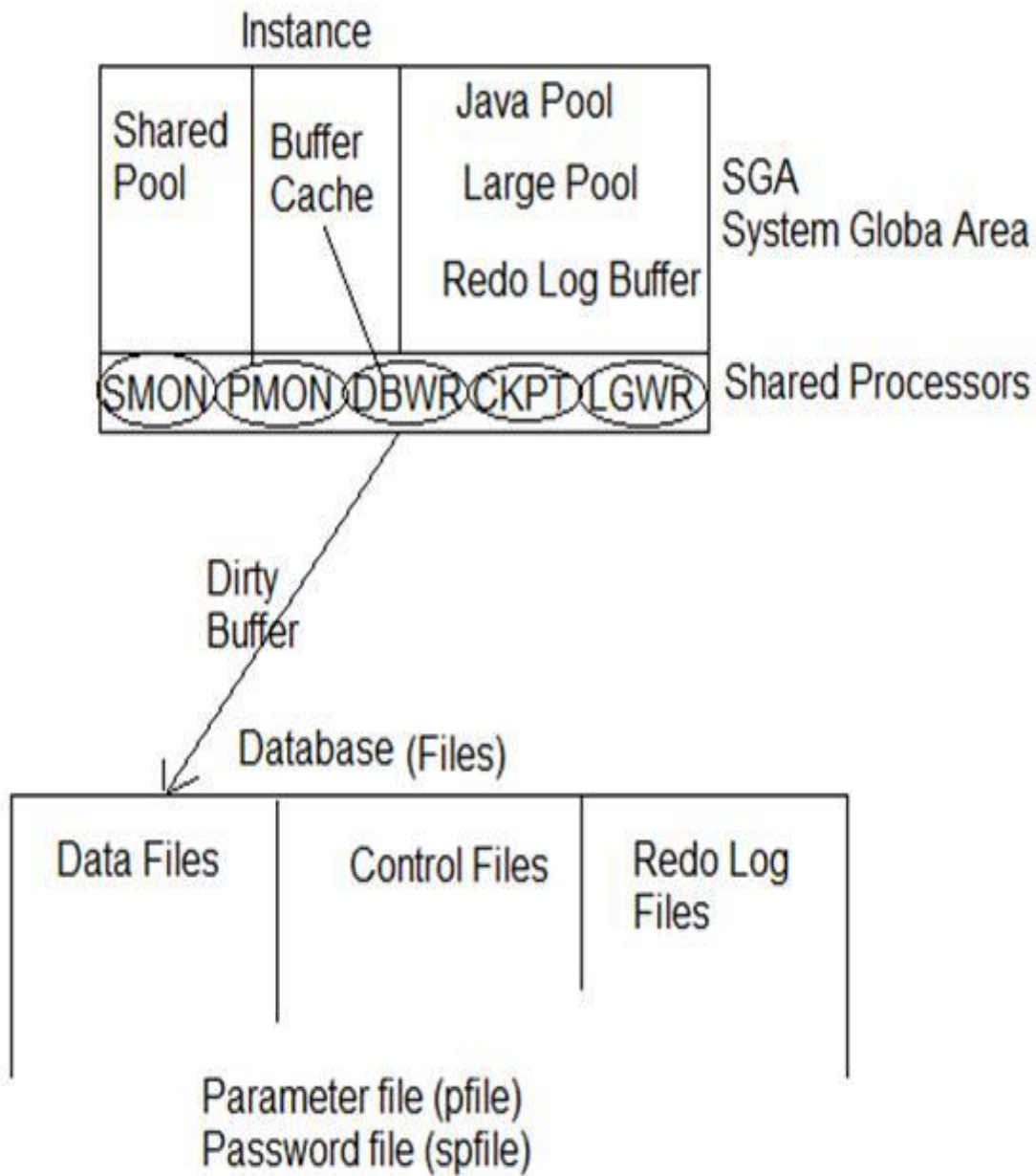
Control Files

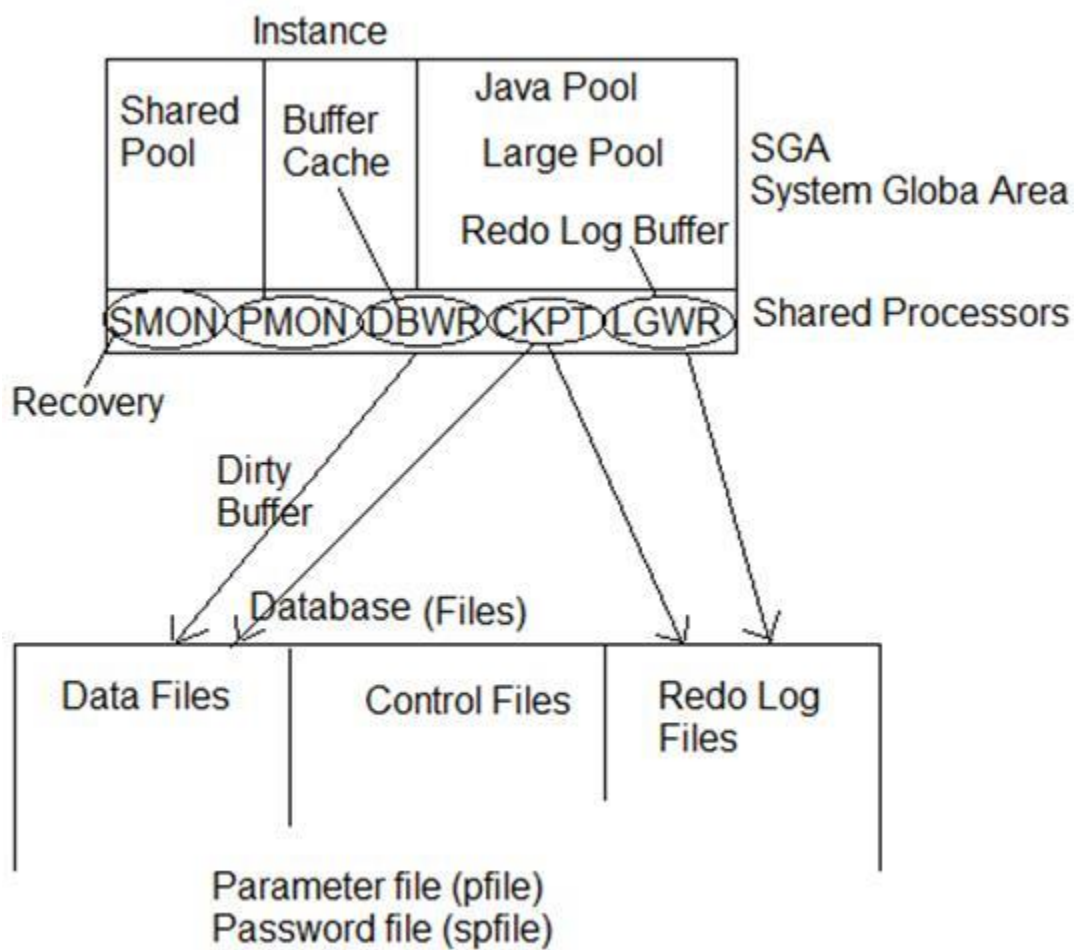
Redo Log
Files

Parameter file (pfile)
Password file (spfile)

Memory Architecture Overview
Overview of SGA & Database

Mahjabeen9782





Overview of SGA & Database

Introduction to Database Administration

TOPIC 52

Configuring Memory

Configuring Memory

Configuring Memory with SGA

Fundamentals of Configuration

Granules

Granule Size

SGA Memory Size

SGA_MAX_SIZE

Setting SGA Target Size

Manual Shared Memory Management

Configuring Memory

Fundamentals of Configuration

Disable automatic memory management of SGA set target and maximum sizes for the SGA
With automatic shared memory management

The database then sets the total size of the SGA to your designated target, and dynamically tunes the sizes of many SGA components.

Configuring Memory

Fundamentals of Configuration

Set the sizes of several individual SGA components with manual shared memory management, thereby determining the overall SGA size. You then manually tune these individual SGA components on an ongoing basis.

Similarly, for the instance PGA, there is automatic/ manual PGA memory management, in which you set a target size for the instance PGA.

Configuring Memory

Granules

SGA memory components include the shared pool (used to allocate memory for SQL and PL/SQL execution), the java pool (used for java objects and other java execution memory), and the buffer cache (used for caching disk blocks).

Configuring Memory

Granules

All SGA components allocate and deallocate space in units of granules.

The granule size is based on the value of the SGA_MAX_SIZE initialization parameter.

Configuring Memory

Granule Size

SGA Memory Amount	Granule Size
Less than or equal to 1 GB	4 MB
Greater than 1 GB and less than or equal to 8 GB	16 MB
Greater than 8 GB and less than or equal to 16 GB	32 MB
Greater than 16 GB and less than or equal to 32 GB	64 MB
Greater than 32 GB and less than or equal to 64 GB	128 MB
Greater than 64 GB and less than or equal to 128 GB	256 MB
Greater than 128 GB	512 MB

Using V\$SGAINFO view to see the granule size that is being used by an instance
Configuring Memory

```
SQL> select name, sum(value)
2  from v$SGA
3  group by name;
```

NAME	SUM(VALUE)	wn.
Database Buffers	352321536	
Redo Buffers	4603904	
Variable Size	713032144	
Fixed Size	1375792	
Total System Global Area	1071333376	bytes
Fixed Size	1375792	bytes
Variable Size	713032144	bytes
Database Buffers	352321536	bytes
Redo Buffers	4603904	bytes

Database mounted.
Database opened.

SGA Memory Size

Configuring Memory

SGA_MAX_SIZE

The SGA_MAX_SIZE initialization parameter specifies the maximum size of the System Global Area for the lifetime of the instance.

If you do not specify SGA_MAX_SIZE, then Oracle Database selects a default value that is the sum of all components specified or defaulted at initialization time.

Configuring Memory

Setting SGA Target Size

You enable the automatic shared memory management feature by setting the SGA_TARGET parameter to a nonzero value. This parameter sets the total size of the SGA.

It replaces the parameters that control the memory allocated for a specific set of individual components, which are now automatically and dynamically resized (tuned) as needed.

Configuring Memory

Setting SGA Target Size

SGA Component	Initialization Parameter
The shared pool	SHARED_POOL_SIZE
The large pool	LARGE_POOL_SIZE
The Java pool	JAVA_POOL_SIZE
The buffer cache	DB_CACHE_SIZE
The Streams pool	STREAMS_POOL_SIZE

Configuring Memory

Setting SGA Target Size

SGA Component	Initialization Parameter
The log buffer	LOG_BUFFER
The keep and recycle buffer caches	DB_KEEP_CACHE_SIZE
	DB_RECYCLE_CACHE_SIZE
Nonstandard block size buffer caches	DB_nK_CACHE_SIZE

In addition to setting SGA_TARGET to a nonzero value, set to zero all initialization parameters of first Table to enable full automatic tuning of the automatically sized SGA components.

Configuring Memory

Manual Shared Memory Management

SHARED_POOL_SIZE = 200M

DB_CACHE_SIZE = 500M

LARGE_POOL_SIZE=200M

Also assume the following query results:

Query	Result
SELECT SUM(value) FROM V\$SGA	1200M
SELECT CURRENT_SIZE FROM V\$SGA_DYNAMIC_FREE_MEMORY	208M
ALTER SYSTEM SET SGA_TARGET = 992M;	
ALTER SYSTEM SET SHARED_POOL_SIZE = 0;	
ALTER SYSTEM SET LARGE_POOL_SIZE = 0;	
ALTER SYSTEM SET JAVA_POOL_SIZE = 0;	
ALTER SYSTEM SET DB_CACHE_SIZE = 0;	
ALTER SYSTEM SET STREAMS_POOL_SIZE = 0;	

where 992M = 1200M minus 208M.

End

Introduction to Database Administration

TOPIC 53

Memory Related Data Dictionary Views

Memory Related Data Dictionary Views

Memory Data Dictionary Views

Memory Parameters

Memory Target & SPFILE

Tune SGA Target

PGA related Queries

Table Size

Blocks & Extents Sizes

Database Block Size

Extent and PCT_FREE Sizes

Maximum Size of Data File

Maximum Size of Data File with Blocks

Memory Related Data Dictionary Views

Memory Parameters

```
SQL> show parameter memory
```

NAME	TYPE	VALUE
hi_shared_memory_address	integer	0
memory_max_target	big integer	1232M
memory_target	big integer	1232M
shared_memory_address	integer	0

```
SQL> show memory_target
```

```
SQL> show parameter SGA
```

NAME	TYPE	VALUE
lock_sga	boolean	FALSE
pre_page_sga	boolean	FALSE
sga_max_size	big integer	1G
sga_target	big integer	0

Memory Related Data Dictionary Views

Memory Parameters

```
SQL> show parameter target
```

NAME	TYPE	VALUE
archive_lag_target	integer	0
db_flashback_retention_target	integer	1440
fast_start_io_target	integer	0
fast_start_mttr_target	integer	0
memory_max_target	big integer	1232M
memory_target	big integer	1232M
parallel_servers_target	integer	16
pga_aggregate_target	big integer	0
sga_target	big integer	0

```
SQL>
```

```
SQL> set linesize 200
```

```
SQL> select substr(name,1,10) fname, substr(value,1,100) fvalue  
2 from v$parameter  
3 where name='spfile';
```

FNAME	FVALUE
spfile	E:\APP\HAIDER\PRODUCT\11.2.0\DBHOME_1\DATABASE\SPFILEORCL.ORA

Memory Related Data Dictionary Views

Memory Target & SPFILE

```
create pfile='E:\app\Haider\admin\orcl\pfile  
\ init.ora.862016235218' from 'E:\app\Haider\product\11.2.0\dbhome_1\dfs\init.ora';
```

```
Alter system set memory_target=1021 M scope=spfile;
```

Memory Related Data Dictionary Views

Memory Target & SPFILE

```
Alter system set SGA_TARGET=0 scope=spfile;  
Alter system set PGA_AGGREGATE_TARGET=0  
scope=spfile;
```

```
shutdown immediate Show parameter memory
```

Memory Related Data Dictionary Views

Tune SGA Target

Desc V\$SGA_TARGET_ADVICE

SGA_SIZE	ESTD_DB_TIME
404	100326
606	97377
808	96681
1010	96333
1212	96323
1414	96323
1616	96323

7 rows selected.

```
select sga_size,estd_db_time from V$SGA_TARGET_ADVICE;
```

Memory Related Data Dictionary Views

PGA related Queries

```
select low_optimal_size/1024 low_kb, (high_optimal_size+1)/1024 high_kb,  
       optimal_executions, onepass_executions, multipasses_executions  
from v$sql_workarea_histogram
```

LOW_KB	HIGH_KB	OPTIMAL_EXECUTIONS	ONEPASS_EXECUTIONS	MULTIPASSES_EXECUTIONS
2	4	354667	0	0
64	128	626	0	0
128	256	190	0	0
256	512	532	0	0
512	1024	3728	0	0
1024	2048	110	0	0
2048	4096	210	0	0
4096	8192	10	0	0
8192	16384	1	0	0

9 rows selected.

```
where total_executions <> 0;
```

Memory Related Data Dictionary Views

Table Size

Login with scott/tiger Desc dba_segments
select sum(bytes)/1024/1024 Table_size_MB
from dba_segments

```
TABLE_SIZE
-----
.0625
```

where segment_name ='EMP';
Memory Related Data Dictionary Views

Blocks & Extents Sizes

select distinct tablespace_name, blocks, extents
from dba_segments;

TABLESPACE_NAME	BLOCKS	EXTENTS
WRK_NOVUS	512	4
SYSTEM	1536	27
PIND_TSDEV	24	3
PIND_TSDEV	128	1
PIND_TSDEV	512	4
WRK_NOVUS	1792	7
SYS_AUX	408	5
UNDOTBS1	144	3
SYSTEM	896	22
SYSTEM	16	2
SYS_AUX	112	14

Memory Related Data Dictionary Views

Database Block Size

Show parameter (to see db_block_size) or
select value
from v\$parameter
where name = 'db_block_size'; Value
8192

Means 8KB block size
Memory Related Data Dictionary Views

Extent and PCT_FREE Sizes

select avg_space, initial_extent, PCT_FREE from ALL_TABLES
where table_name = 'EMP';

AVG_SPACE	INITIAL_EXTENT	PCT_FREE
0	65536	10

Memory Related Data Dictionary Views

Maximum Size of Data File

How to see block size and max file size with DBA_DATA_FILES (4 Millions data blocks)

Find db block size from v\$parameter 4194304 * db_block_size

Memory Related Data Dictionary Views

Maximum Size of Data File

If your db_block_size is 8kb, then your maximum datafile size is:

$4194304 * 8192 = 34359738368 / 1024$
 $/ 1024 / 1024 = 32GB.$

select file_name, tablespace_name, blocks
from dba_data_files;

Memory Related Data Dictionary Views

Each Oracle datafile can contain maximum 4194303 (4 Million) data blocks.

Block Size | Maximum Datafile Size

2k 4194303 * 2k = 8 GB

4k 4194303 * 4k = 16 GB

8k 4194303 * 8k = 32 GB

16k 4194303 * 16k = 64 GB

32k 4194303 * 32k = 128 GB

In Oracle Database 10g, BIGFILE tablespace was introduced.
(4 billion) data blocks.

Block Size | Maximum Datafile Size

2k 4294967295 * 2k = 8 TB

4k 4294967295 * 4k = 16 TB

8k 4294967295 * 8k = 32 TB

16k 4294967295 * 16k = 64 TB

32k 4294967295 * 32k = 128 TB

Maximum Size of Data File with Blocks

End

Introduction to Database Administration

TOPIC 54

Backup using MS SQL Server

Backup using MS SQL Server

Backup using MS SQL Server

Terminologies

Failures of Database

Create a Full Backup

Create a Simple test database

Backup using MSSMS

Steps of Backup using MSSMS

Backup on non-default Location

Backup using MS SQL Server

Terminologies

Back up [verb]

The process of creating a backup [noun] by copying data records from a SQL Server database, or log records from its transaction log.

Backup [noun]

A copy of data that can be used to restore and recover the data after a failure. Backups of a database can also be used to restore a copy the database to a new location.

Backup using MS SQL Server

Terminologies

Backup device or media

A disk or tape device to which SQL Server backups are written and from which they can be restored.

SQL Server backups can also be written to an Azure Blob storage service, and URL format is used to specify the destination and the name of the backup file

Backup using MS SQL Server

Terminologies

Full backup

A data backup that contains all the data in a specific database or set of file groups or files, and also enough log to allow for recovering that data.

Backup using MS SQL Server

Failures of Database

Media failure.

User errors, for example, dropping a table by mistake.

Hardware failures, for example, a damaged disk drive or permanent loss of a server.

Backup using MS SQL Server

Failures of Database

Natural disasters. By using SQL Server Backup to Azure Blob storage service, you can create an off-site backup in a different region than your on- premises location, to use in the event of a natural disaster affecting your on-premises location.

Backup using MS SQL Server

Create a Full Backup

Security

TRUSTWORTHY is set to OFF on a database backup

Permission

BACKUP DATABASE and BACKUP

LOG permissions default to members of the sysadmin fixed server role and the db_owner and db_backupoperator fixed database roles.

Backup using MS SQL Server

Create a Sample Test Database

```
USE [master]
GO

CREATE DATABASE [SQLTestDB]
GO

USE [SQLTestDB]
GO
CREATE TABLE SQLTest
(
    ID INT NOT NULL PRIMARY KEY,
    c1 VARCHAR(100) NOT NULL,
    dt1 DATETIME NOT NULL DEFAULT getdate()
);
GO

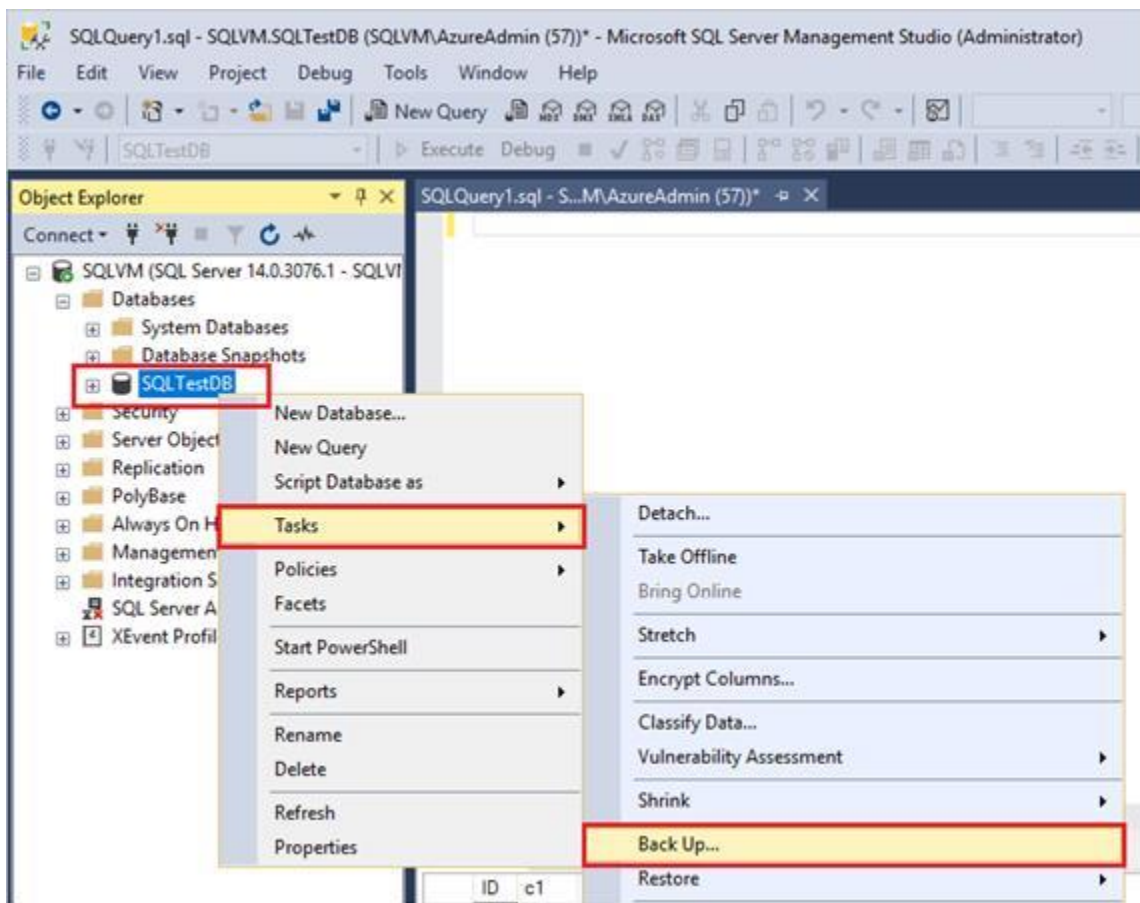
USE [SQLTestDB]
GO

INSERT INTO SQLTest (ID, c1) VALUES (1, 'test1')
INSERT INTO SQLTest (ID, c1) VALUES (2, 'test2')
INSERT INTO SQLTest (ID, c1) VALUES (3, 'test3')
INSERT INTO SQLTest (ID, c1) VALUES (4, 'test4')
INSERT INTO SQLTest (ID, c1) VALUES (5, 'test5')
GO

SELECT * FROM SQLTest
GO
```

Backup using MS SQL Server

Back up using MSSMS



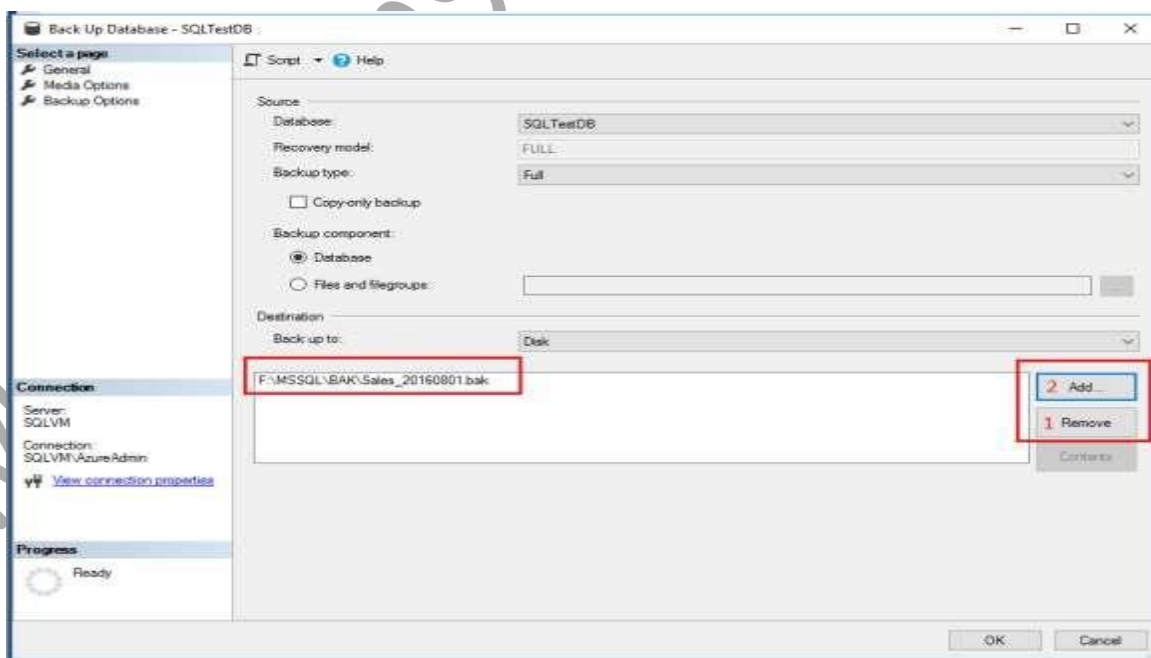
Backup using MS SQL Server
Steps of Back up using MSSMS

In this example, the `SQLTestDB` database will be backed up to disk at a location of your choice.

1. After connecting to the appropriate instance of the Microsoft SQL Server Database Engine, in **Object Explorer**, expand the server tree.
2. Expand **Databases**, right-click `SQLTestDB`, point to **Tasks**, and then click **Back Up....**
3. On the **General** page in the **Destination** section select **Disk** from the **Back up to:** drop-down list.
4. Select **Remove** until all existing backup files have been removed.
5. Select **Add** and the **Select Backup Destination** dialog box will open.
6. Enter a valid path and file name in the **File name** text box and use **.bak** as the extension to simplify the classification of this file.
7. Click **OK** and then click **OK** again to initiate the backup.
8. When the backup completes successfully, click **OK** to close the SQL Server Management Studio dialog box.

Backup using MS SQL Server

Back up on non-default Location



TOPIC 55

What are Control Files & Creating Control Files

Creating Control Files

Control Files

What is Control Files?

Managing Control Files

Back up Control Files

Creating Control Files

Multiplex Copying of Control Files

Creating Control Files

What is Control Files?

Control file contains

The database name

Names and locations of associated data files and redo log files

The timestamp of the database creation

The current log sequence number

Checkpoint information

Creating Control Files

Managing Control Files?

DB server writes into control file when DB is open.

DB is mounted for recovery with control file.

More control files can be created

Creating Control Files

Back up Control Files?

Adding, dropping, or renaming datafiles

Adding or dropping a tablespace, or altering the read/write state of the tablespace

Adding or dropping redo log files or groups

Creating Control Files

Creating Control Files?

C > Stuff (E:) > app > Haider > oradata > orcl >

Name	Date modified
prod	19-Apr-17
CONTROL01.CTL	15-Feb-20
IND_MYDEV01.DBF	15-Feb-20
IND_SDEV01.DBF	15-Feb-20
REDO01.LOG	15-Feb-20
REDO02.LOG	15-Feb-20
REDO03.LOG	15-Feb-20
SDEV01.DBF	15-Feb-20
SYSAUX01.DBF	15-Feb-20
SYSTEM01.DBF	15-Feb-20
TEMP01.DBF	15-Feb-20
UNDOTBS01.DBF	15-Feb-20
USERS01.DBF	15-Feb-20

Creating Initial Control Files

Creating Additional Copies, Renaming, and Relocating Control Files

Creating New Control Files

Initial Control Files are

CONTROL_FILES =
(/u01/oracle/prod/control01.ctl,
/u02/oracle/prod/control02.ctl,
/u03/oracle/prod/control03.ctl)

Creating Control Files

Multiplex copying of Control Files

To add a multiplexed copy of the current control file or to rename a control file:

Shut down the database.

Copy an existing control file to a new location, using operating system commands.

Edit the CONTROL_FILES parameter in the database initialization parameter file to add the new control file name, or to change the existing control filename.

Restart the database.

Introduction to Database Administration

TOPIC 56

Back up Control Files

Back up Control Files

Back up and New Control Files

When to Create Control Files?

Create Control File Statement

Data Dictionary Views for Control Files?

Using Data Dictionary Views with user sys

SQL of V\$LOGFILE

SQL of V\$DATAFILE

SQL of V\$CONTROLFILE

SQL of V\$PARAMETER

Backing up Control Files?

Steps in creating new Control Files?

Back up Control Files

When to Create Control Files?

All control files for the database have been permanently damaged and you do not have a control file backup.

You want to change the database name.

Back up Control Files

Create Control File Statement

```

CREATE CONTROLFILE
SET DATABASE prod
LOGFILE GROUP 1 ('/u01/oracle/prod/redo01_01.log',
                '/u01/oracle/prod/redo01_02.log'),
GROUP 2 ('/u01/oracle/prod/redo02_01.log',
        '/u01/oracle/prod/redo02_02.log'),
GROUP 3 ('/u01/oracle/prod/redo03_01.log',
        '/u01/oracle/prod/redo03_02.log')

RESETLOGS
DATAFILE '/u01/oracle/prod/system01.dbf' SIZE 3M,
        '/u01/oracle/prod/rbs01.dbs' SIZE 5M,
        '/u01/oracle/prod/users01.dbs' SIZE 5M,
        '/u01/oracle/prod/temp01.dbs' SIZE 5M
MAXLOGFILES 50
MAXLOGMEMBERS 3
MAXLOGHISTORY 400
MAXDATAFILES 200
MAXINSTANCES 6
ARCHIVELOG;

```

Back up Control Files

Data Dictionary Views for Control Files?

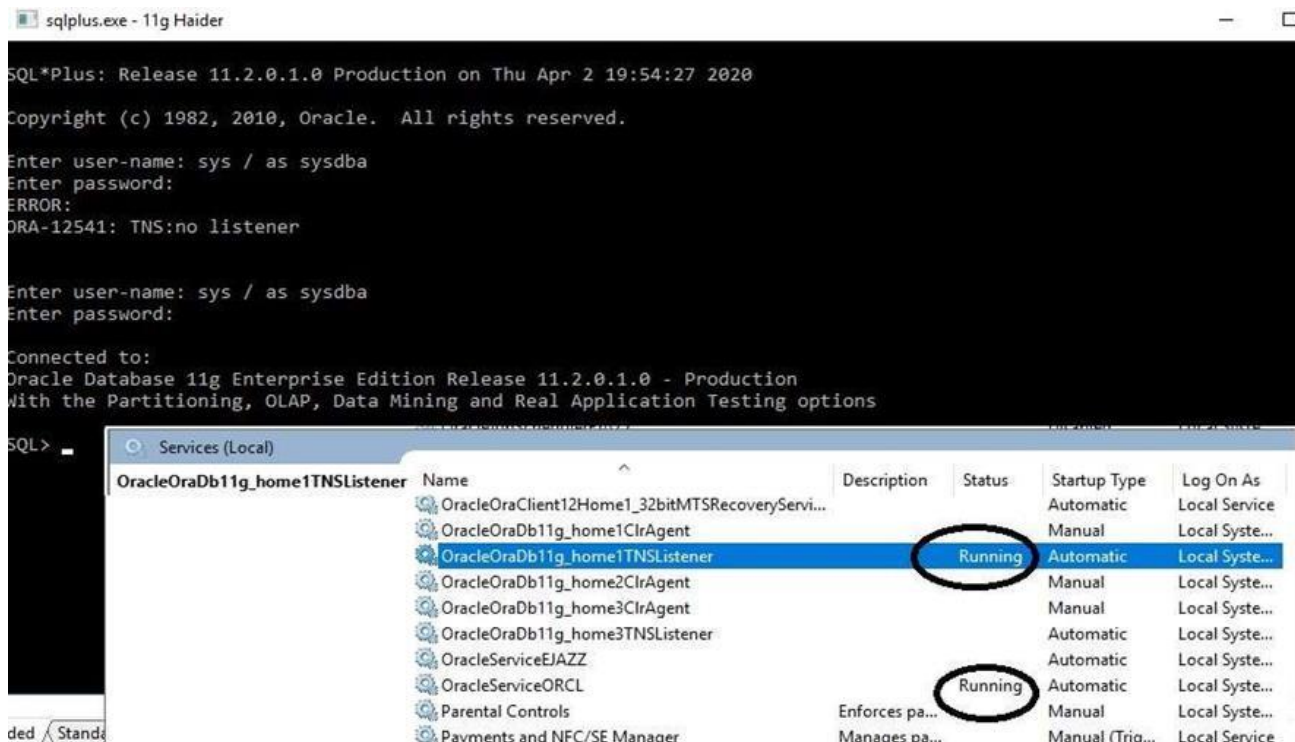
SELECT MEMBER FROM V\$LOGFILE;

SELECT NAME FROM V\$DATAFILE;

SELECT VALUE FROM V\$PARAMETER WHERE NAME = 'control_files'

Back up Control Files

Using Data Dictionary Views with user sys



Back up Control Files
SQL of V\$LOGFILE

```
SQL> SELECT MEMBER FROM V$LOGFILE;
```

MEMBER
E:\APP\HAIDER\ORADATA\ORCL\REDO003.LOG
E:\APP\HAIDER\ORADATA\ORCL\REDO002.LOG
E:\APP\HAIDER\ORADATA\ORCL\REDO001.LOG

```
SQL>
```

Back up Control Files
SQL of V\$DATAFILE

```

SQL> SELECT NAME FROM V$DATAFILE;

NAME
-----
E:\APP\HAIDER\ORADATA\ORCL\SYSTEM01.DBF
E:\APP\HAIDER\ORADATA\ORCL\SYSAUX01.DBF
E:\APP\HAIDER\ORADATA\ORCL\UNDOTBS01.DBF
E:\APP\HAIDER\ORADATA\ORCL\USERS01.DBF
E:\APP\HAIDER\ORADATA\ORCL\IND_MYDEV01.DBF
E:\APP\HAIDER\ORADATA\ORCL\SDEV01.DBF
E:\APP\HAIDER\ORADATA\ORCL\IND_SDEV01.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PDEV01.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PDEV02.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PDEV03.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PIND_DEV.DBF

NAME
-----
E:\APP\HAIDER\ORADATA\ORCL\PROD\PTSIMG01.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PTSIMG02.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PTSIMG03.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PTSIMG04.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PSTUCRSREG01.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PSTUCRSREG02.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PEMARKS01.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\PEMARKS02.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\TSIMG01.DBF
E:\APP\HAIDER\ORADATA\ORCL\PROD\TSIMG02.DBF
E:\APP\HAIDER\PRODUCT\11.2.0\DBHOME_1\ORADATA\NOVUS01.DBF

```

Back up Control Files

SQL of V\$CONTROLFILE

With default installation, oracle contains only two control files

```

SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
-----
E:\APP\HAIDER\ORADATA\ORCL\CONTROL01.CTL
E:\APP\HAIDER\FLASH_RECOVERY_AREA\ORCL\CONTROL02.CTL

SQL>

```

Back up Control Files

SQL of V\$PARAMETER

```
SQL> show parameter control_files
```

NAME	TYPE	VALUE
control_files	string	E:\APP\HAID TROL01.CTL, H_RECOVERY_ .CTL \\CONTROL02.CTL

Back up Control Files

Backing up Control Files?

Back up the control file to a binary file (duplicate of existing control file) using the following statement:

```
ALTER DATABASE BACKUP  
CONTROLFILE TO  
'/oracle/backup/control.bkp';
```

Produce SQL statements that can later

be used to re-create your control file: ALTER DATABASE BACKUP CONTROLFILE TO TRACE;

Back up Control Files

Steps in creating new Control Files?

Shut down the database.

If the database is open, shut down the database normally if possible. Use the IMMEDIATE or ABORT clauses only as a last resort.

Back up all data files and redo log files of the database.

Back up Control Files

Steps in creating new Control Files?

Start up a new instance, but do not mount or open the database: **STARTUP NOMOUNT**
Create a new control file for the database using the CREATE CONTROLFILE statement.
Back up Control Files

Executing steps in creating new Control Files?

login sqlplus with user: sys / as sysdba select name from v\$controlfile;
Alter database backup controlfile to trace as 'E:\APP\HAIDER\ORADATA\ORCL\control.
bkp';
Back up Control Files

Executing steps in creating new Control Files?

alter system set control_files='E:\APP\HAIDER\ORADAT A\ORCL\control01.ctl',
'E:\APP\HAIDER\FLASH_RECOVERY_AR
EA\ORCL\control02.ctl', 'E:\APP\HAIDER\ORADATA\ORCL\contr ol03.ctl' scope=spfile;
shutdown immediate exit
Back up Control Files

Executing steps in creating new Control Files?

Use copy-past file in windows file explorer or use DOS command
E:\APP\HAIDER\ORADATA\ORCL>copy
control01.ctl control03.ctl
Back up Control Files

Executing steps in creating new Control Files?

sqlplus
Enter user-name: sys / as sysdba Enter password:
Connected to an idle instance.

SQL> startup
ORACLE instance started. shutdown immediate exit
Back up Control Files

Executing steps in creating new Control Files?

SQL>select name from v\$controlfile; NAME
E:\APP\HAIDER\ORADATA\ORCL\CONTROL 01.CTL
E:\APP\HAIDER\FLASH_RECOVERY_AREA\ ORCL\CONTROL02.CTL
E:\APP\HAIDER\ORADATA\ORCL\CONTROL 03.CTL

Back up Control Files

Executing steps in creating new Control Files?

is PC > Stuff (E:) > app > Haider > oradata > orcl

Name	Date modified	Type	Size
prod	19-Apr-17 5:51 PM	File folder	
CONTROL.BKP	26-May-20 11:09 ...	BKP File	8 KB
CONTROL01.CTL	26-May-20 11:14 ...	Visual Basic U...	9,520 KB
CONTROL03.CTL	26-May-20 11:14 ...	Visual Basic U...	9,520 KB
IND_MYDEV01.DBF	26-May-20 11:09 ...	DBF File	1,024,008 KB
IND_SDEV01.DBF	26-May-20 11:09 ...	DBF File	512,008 KB
REDO01.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
REDO02.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
REDO03.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
SDEV01.DBF	26-May-20 11:09 ...	DBF File	1,024,008 KB
SYS_AUX01.DBF	26-May-20 11:09 ...	DBF File	706,568 KB
SYSTEM01.DBF	26-May-20 11:09 ...	DBF File	727,048 KB
TEMP01.DBF	26-May-20 10:20 ...	DBF File	116,744 KB
UNDOTBS01.DBF	26-May-20 11:09 ...	DBF File	312,328 KB
USERS01.DBF	26-May-20 11:09 ...	DBF File	6,408 KB

Introduction to Database Administration

TOPIC 57

Trouble Shouting Control Files

Trouble Shouting Control Files

Trouble Shouting Control Files

SQL of V\$CONTROLFILE

Executing steps in creating new Control Files

Steps in creating new Control Files

Executing Commands on SQL

Executing steps in creating new Control Files
Backup Control Files using Startup Mount
Trouble Shouting Control Files

Trouble Shouting Control Files

Alter System Set Command & initialization File
Executing steps in creating new
Trouble Shouting Control Files

Trouble Shouting Control Files

Control Files using SQL
Use of Database MOUNT & NO MOUNT
Database is in normal mode (consistent database)
Executing steps in creating new Control Files
Warning for storing Control Files
Trouble Shouting Control Files

When to Create Control Files?

All control files for the database have been permanently damaged and you do not have a control file backup.
You want to change the database name.
Trouble Shouting Control Files

SQL of V\$CONTROLFILE

With default installation, oracle contains only two control files

```
SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
-----
E:\APP\HAIDER\ORADATA\ORCL\CONTROL01.CTL
E:\APP\HAIDER\FLASH_RECOVERY_AREA\ORCL\CONTROL02.CTL

SQL>
```

Trouble Shouting Control Files

is PC > Stuff (E:) > app > Haider > oradata > orcl

Name	Date modified	Type	Size
prod	19-Apr-17 5:51 PM	File folder	
CONTROL.BKP	26-May-20 11:09 ...	BKP File	8 KB
CONTROL01.CTL	26-May-20 11:14 ...	Visual Basic U...	9,520 KB
CONTROL03.CTL	26-May-20 11:14 ...	Visual Basic U...	9,520 KB
IND_MYDEV01.DBF	26-May-20 11:09 ...	DBF File	1,024,008 KB
IND_SDEV01.DBF	26-May-20 11:09 ...	DBF File	512,008 KB
REDO01.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
REDO02.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
REDO03.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
SDEV01.DBF	26-May-20 11:09 ...	DBF File	1,024,008 KB
SYSAUX01.DBF	26-May-20 11:09 ...	DBF File	706,568 KB
SYSTEM01.DBF	26-May-20 11:09 ...	DBF File	727,048 KB
TEMP01.DBF	26-May-20 10:20 ...	DBF File	116,744 KB
UNDOTBS01.DBF	26-May-20 11:09 ...	DBF File	312,328 KB
USERS01.DBF	26-May-20 11:09 ...	DBF File	6,408 KB

Executing steps in creating new Control Files

Trouble Shouting Control Files

Steps in creating new Control Files

Shut down the database.

If the database is open, shut down the database normally if possible. Use the IMMEDIATE or ABORT clauses only as a last resort.

Back up all data files and redo log files of the database.

Trouble Shouting Control Files

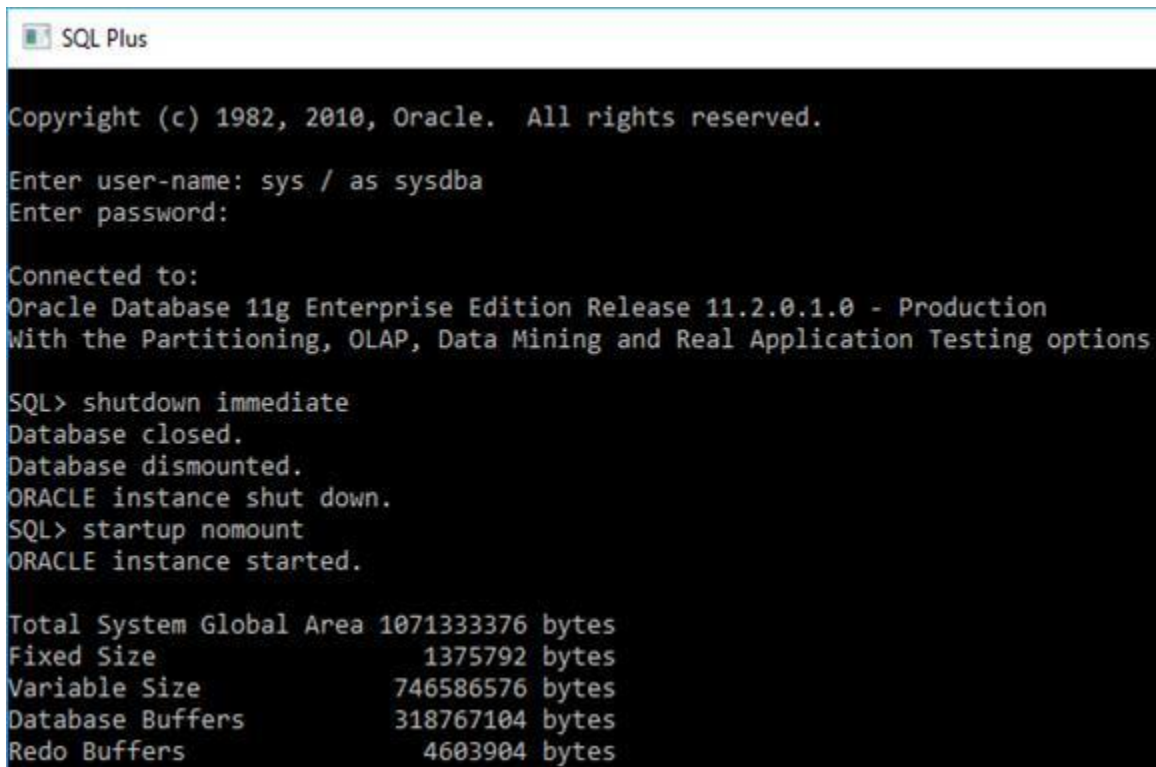
Steps in creating new Control Files

Start up a new instance, but do not mount or open the database: **STARTUP NOMOUNT**

Create a new control file for the database using the CREATE CONTROLFILE statement.

Trouble Shouting Control Files

Executing Commands on SQL



```
SQL Plus

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: sys / as sysdba
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup nomount
ORACLE instance started.

Total System Global Area 1071333376 bytes
Fixed Size 1375792 bytes
Variable Size 746586576 bytes
Database Buffers 318767104 bytes
Redo Buffers 4603904 bytes
```

Trouble Shouting Control Files

Executing steps in creating new Control Files

login sqlplus with user: sys / as sysdba select name from v\$controlfile;
Alter database backup controlfile to trace as 'E:\APP\HAIDER\ORADATA\ORCL\control.b
kp';

Trouble Shouting Control Files

Executing steps in creating new Control Files

Alter system set control_files='E:\APP\HAIDER\ORADAT A\ORCL\control01.ctl',
'E:\APP\HAIDER\FLASH_RECOVERY_AR
EA\ORCL\control02.ctl', 'E:\APP\HAIDER\ORADATA\ORCL\contr ol03.ctl' scope=spfile;
shutdown immediate exit

Trouble Shouting Control Files

Backup Control Files using Startup Mount

```

SQL> shutdown immediate
ORA-01507: database not mounted

ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area 1071333376 bytes
Fixed Size 1375792 bytes
Variable Size 746586576 bytes
Database Buffers 318767104 bytes
Redo Buffers 4603904 bytes
Database mounted.
SQL> Alter database backup controlfile to trace as 'E:\APP\HAIDER\ORADATA\ORCL\control.bkp';
Alter database backup controlfile to trace as 'E:\APP\HAIDER\ORADATA\ORCL\control.bkp'
*
ERROR at line 1:
ORA-01277: file 'E:\APP\HAIDER\ORADATA\ORCL\CONTROL.BKP' already exists

SQL>
SQL> Alter database backup controlfile to trace as 'E:\APP\HAIDER\ORADATA\ORCL\control1.bkp';
Database altered.

```

Trouble Shouting Control Files

Alter System Set Command & initialization File

This PC > Stuff (E:) > app > Haider > flash_recovery_area > orcl

Name	Date modified
ARCHIVELOG	03-Aug-20 10:27 P...
ONLINELOG	07-Sep-16 11:51 AM
CONTROL02.CTL	03-Aug-20 10:31 P...

This PC > Stuff (E:) > app > Haider > admin > orcl > pfile

Name	Date modified
init.ora.862016235218	07-Sep-16 11:50 AM

Trouble Shouting Control Files

Executing steps in creating new Control Files using SQL

Use copy-past file in windows file explorer or use DOS command

```
E:\APP\HAIDER\ORADATA\ORCL>copy
```

```
control01.ctl control03.ctl
```

Trouble Shouting Control Files

Executing steps in creating new Control Files using SQL

```
sqlplus
```

```
Enter user-name: sys / as sysdba Enter password:
```

```
Connected to an idle instance.
```

```
SQL> startup
```

```
ORACLE instance started. shutdown immediate exit
```

Trouble Shouting Control Files

Use of Database MOUNT & NO MOUNT

In NOMOUNT, Backup of existing control files and redo log files is required

In MOUNT, run Alter database backup ...

Trouble Shouting Control Files

Database is in normal mode (consistent database)

```
SQL> shutdown immediate
ORA-01109: database not open
```

```
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.
```

```
Total System Global Area 1071333376 bytes
Fixed Size                  1375792 bytes
Variable Size               746586576 bytes
Database Buffers            318767104 bytes
Redo Buffers                 4603904 bytes
Database mounted.
Database opened.
SQL>
```

Trouble Shouting Control Files

Executing steps in creating new Control Files

```
SQL>select name from v$controlfile;
NAME
```

```
-----
E:\APP\HAIDER\ORADATA\ORCL\CONTROL 01.CTL
E:\APP\HAIDER\FLASH_RECOVERY_AREA\ORCL\CONTROL02.CTL
E:\APP\HAIDER\ORADATA\ORCL\CONTROL 03.CTL
```

Trouble Shouting Control Files

Warning for Storing Control Files

You may want to ensure that control files are created on separate physical devices

End

Introduction to Database Administration

TOPIC 58

Recover & Dropping Control Files

Recover & Dropping Control Files

Recover & Dropping Control Files

Checking for Missing or Extra Files
Handling Errors During CREATE CONTROLFILE
ORA Error - A Sample
Different Locations of Control Files
Executing Commands on SQL
Backing up Control Files
Recover & Dropping Control Files

Recover & Dropping Control Files

Viewing Alert Log using Data Dictionary
Viewing Alert Log using Enterprise Manager
Finding Trace Files
Recovery from Permanent Media Failure
Dropping Control Files
Recover & Dropping Control Files

Checking for Missing or Extra Files

Dropped control files are still shown through data dictionary.
MISSINGnnnn is a flag in control file, gives status of file when seen by the data dictionary but actually does not exist on disk.
Recover & Dropping Control Files

Checking for Missing or Extra Files

If the actual data file corresponding to MISSINGnnnn is read-only or offline normal, then you can make the data file accessible by renaming MISSINGnnnn to the name of the actual data file.
Recover & Dropping Control Files

Checking for Missing or Extra Files

If MISSINGnnnn corresponds to a data file that was not read-only or offline normal, then you cannot use the rename operation to make the data file accessible, because the data file requires media recovery that is precluded by the results of RESETLOGS.

Recover & Dropping Control Files

Checking for Missing or Extra Files

In this case, you must drop the tablespace containing the data file.

Conversely, if a data file listed in the control file is not present in the data dictionary, then the database removes references to it from the new control file. Refers to Alert log in any case.

Recover & Dropping Control Files

Handling Errors During CREATE CONTROLFILE

If Oracle Database sends you an error (usually error ORA-01173, ORA-01176, ORA-01177, ORA-01215, or ORA-01216)

when you attempt to mount and open the database after creating a new control file, the most likely cause is that you omitted a file from the CREATE CONTROLFILE statement or included one that should not have been listed.

Recover & Dropping Control Files

ORA Error – A Sample

Oracle Error: ORA-01173

Error Description:

Data dictionary indicates missing data file from system tablespace

Error Cause:

Either the database has been recovered to a point in time in the future of the control file or a datafile from the system tablespace was omitted from the create control file command previously issued.

Recover & Dropping Control Files

ORA Error – A Sample

Action:

For the former problem you need to recover the database from a more recent control file. For the later problem, simply recreate the control file checking to be sure that you include all the datafiles in the system tablespace.

Recover & Dropping Control Files

Different Locations of Control Files

is PC > Stuff (E:) > app > Haider > oradata > orcl

Name	Date modified	Type	Size
prod	19-Apr-17 5:51 PM	File folder	
CONTROL.BKP	26-May-20 11:09 ...	BKP File	8 KB
CONTROL01.CTL	26-May-20 11:14 ...	Visual Basic U...	9,520 KB
CONTROL03.CTL	26-May-20 11:14 ...	Visual Basic U...	9,520 KB
IND_MYDEV01.DBF	26-May-20 11:09 ...	DBF File	1,024,008 KB
IND_SDEV01.DBF	26-May-20 11:09 ...	DBF File	512,008 KB
REDO01.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
REDO02.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
REDO03.LOG	26-May-20 11:09 ...	Text Document	51,201 KB
SDEV01.DBF	26-May-20 11:09 ...	DBF File	1,024,008 KB
SYSAUX01.DBF	26-May-20 11:09 ...	DBF File	706,568 KB
SYSTEM01.DBF	26-May-20 11:09 ...	DBF File	727,048 KB
TEMP01.DBF	26-May-20 10:20 ...	DBF File	116,744 KB
UNDOTBS01.DBF	26-May-20 11:09 ...	DBF File	312,328 KB
USERS01.DBF	26-May-20 11:09 ...	DBF File	6,408 KB

Recover & Dropping Control Files

Executing Commands on SQL

```
SQL Plus

Copyright (c) 1982, 2010, Oracle. All rights reserved.

Enter user-name: sys / as sysdba
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup nomount
ORACLE instance started.

Total System Global Area 1071333376 bytes
Fixed Size 1375792 bytes
Variable Size 746586576 bytes
Database Buffers 318767104 bytes
Redo Buffers 4603904 bytes
```

Recover & Dropping Control Files

Backing up Control Files

login sqlplus with user: sys / as sysdba select name from v\$controlfile;

Alter database backup controlfile to

trace as 'E:\APP\HAIDER\ORADATA\ORCL\control.bkp';

Recover & Dropping Control Files

Backing up Control Files

Produce SQL statements that can later be used to re-create your control file:

ALTER DATABASE BACKUP CONTROLFILE TO TRACE;

Recover & Dropping Control Files

Backing up Control Files

This command writes a SQL script to a trace file where it can be captured and edited to reproduce the control file.

View the **alert log** to determine the name and location of the trace file.

Recover & Dropping Control Files

Viewing Alert Log using Data Dictionary

```
SQL> SELECT substr(Inst_id,1, 3),
2  substr(name, 1, 25) Diag_name,
3  substr(value, 1, 100) Diag_value
4  from V$DIAG_INFO;
```

SUB	DIAG_NAME	DIAG_VALUE
1	Diag Enabled	TRUE
1	ADR Base	e:\app\haider
1	ADR Home	e:\app\haider\diag\rdbms\orcl\orcl
1	Diag Trace	e:\app\haider\diag\rdbms\orcl\orcl\trace
1	Diag Alert	e:\app\haider\diag\rdbms\orcl\orcl>alert
1	Diag Incident	e:\app\haider\diag\rdbms\orcl\orcl\incident
1	Diag Cdump	e:\app\haider\diag\rdbms\orcl\orcl\cdump
1	Health Monitor	e:\app\haider\diag\rdbms\orcl\orcl\hm
1	Default Trace File	e:\app\haider\diag\rdbms\orcl\orcl\trace\orcl_ora_25104.trc
1	Active Problem Count	0
1	Active Incident Count	0

11 rows selected.

Recover & Dropping Control Files

Viewing Alert Log using Enterprise Manager

Access the Database Home page in Enterprise Manager.
Under Related Links, click Alert Log Contents.
The View Alert Log Contents page appears.
Select the number of entries to view, and then click Go
Recover & Dropping Control Files

Finding Trace Files

Trace files are stored in the Automatic Diagnostic Repository (ADR), in the trace directory under each ADR home.

To find the trace file for your current session:

```
SELECT VALUE FROM V$DIAG_INFO  
WHERE NAME = 'Default Trace File';
```

Recover & Dropping Control Files

Finding Trace Files

To find all trace files for the current instance:

```
SELECT VALUE FROM V$DIAG_INFO  
WHERE NAME = 'Diag Trace';
```

The path to the ADR trace directory for the current instance is returned.

Recover & Dropping Control Files

Finding Trace Files

To determine the trace file for each Oracle Database process:

```
SELECT PID, PROGRAM, TRACEFILE FROM V$PROCESS;
```

```
SQL> SELECT VALUE FROM V$DIAG_INFO WHERE NAME = 'Diag Trace';
```

VALUE
e:\app\haider\diag\rdbms\orcl\orcl\trace

Name	Date modified
alert_orcl.log	04-Aug-20 10:45 A...
orcl_cjq0_5516.trc	02-Aug-20 6:21 PM
orcl_cjq0_5516.trm	02-Aug-20 6:21 PM
orcl_cjq0_14276.trc	11-Jul-20 10:01 AM
orcl_cjq0_14276.trm	11-Jul-20 10:01 AM
orcl_cjq0_26512.trc	04-Aug-20 10:44 A...

Recover & Dropping Control Files

Recovery from Permanent Media Failure

With the instance shut down, use an operating system command of copy

```
% cp /u01/oracle/prod/control01.ctl
```

```
/u04/oracle/prod/control03.ctl
```

Recover & Dropping Control Files

Recovery from Permanent Media Failure

Edit the CONTROL_FILES parameter in the initialization parameter file to replace the bad location with the new location:

```
CONTROL_FILES =
```

```
(/u01/oracle/prod/control01.ctl,
```

```
/u02/oracle/prod/control02.ctl,
```

```
/u04/oracle/prod/control03.ctl)
```

```
SQL> STARTUP
```

Recover & Dropping Control Files

Dropping Control Files

Database should have at least two control files at all times.

Shut down the database.

Edit the CONTROL_FILES parameter in the database initialization parameter file to delete the old control file name.

Restart the database.

End

Introduction to Database Administration

TOPIC 59

What are Redo Log Files?

What are Redo Log Files?

Redo Log Files?

What is the Redo Log?

Redo Threads

Redo Log Contents

Writing to Redo Log Files

Active & In-active Redo Log Files

Overwriting Redo Log Files

What are Redo Log Files?

What is the Redo Log?

The most crucial structure for recovery operations is the redo log

The structure consists of two or more pre-allocated files that store all changes made to the database as they occur.

What are Redo Log Files?

What is the Redo Log?

Every instance of an Oracle Database has an associated redo log to protect the database in case of an instance failure.

Redo Log files stores all the change information for the database and are used by Oracle during Database recovery

What are Redo Log Files?

Redo Threads?

When speaking in the context of multiple database instances, the redo log for each database instance is also referred to as a redo thread.

In typical configurations, only one database instance accesses an Oracle Database, so only one thread is present.

What are Redo Log Files?

Redo Log Contents?

Redo log files are filled with redo records or redo entry, is made up of a group of change vectors, each of which is a description of a change made to a single block in the database.

What are Redo Log Files?

Redo Log Contents?

For example, if you change a salary value in an employee table, you generate a redo record containing change vectors that describe changes to the data segment block for the table, the undo segment data block, and the transaction table of the undo segments.

Redo records are buffered in a circular fashion in the redo log buffer of the SGA.

What are Redo Log Files?

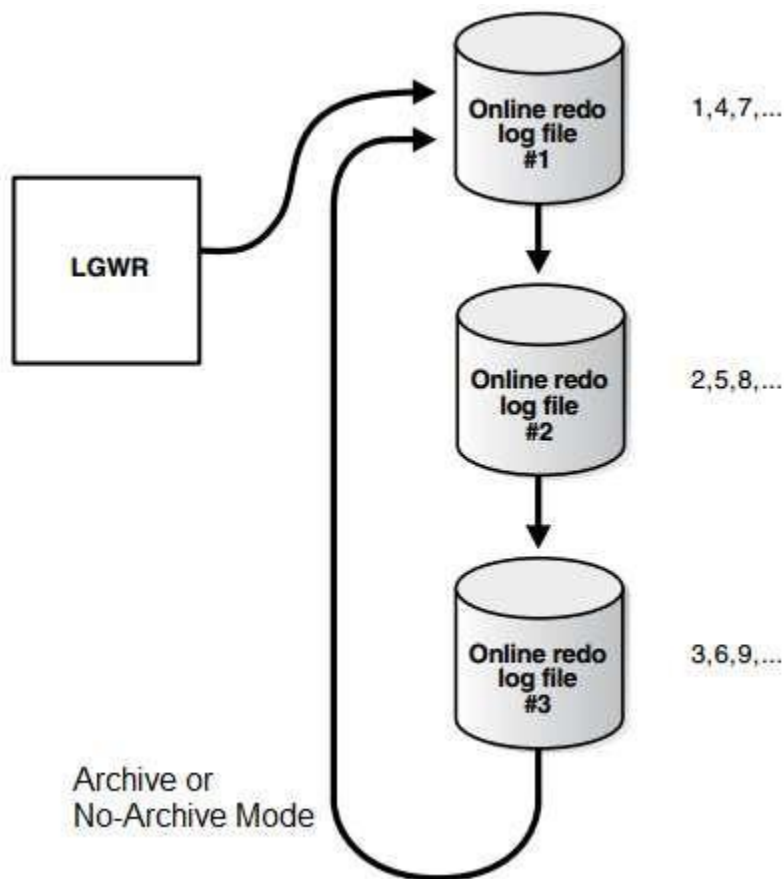
Redo Log Contents?

Redo records are written to one of the redo log files by the Log Writer (LGWR) database background process.

Whenever a transaction is committed, LGWR writes the transaction redo records from the redo log buffer of the SGA to a redo log file, and assigns a system change number (SCN) to identify the redo records for each committed transaction.

Transactions are committed when redo records are written to the disk safely.

What are Redo Log Files?



Writing to Redo Log files

LGWR writes to redo log files in a circular fashion. When the current redo log file fills, LGWR begins writing to the next available redo log file. When the last available redo log file is filled, LGWR returns to the first redo log file and writes to it, starting the cycle again.

What are Redo Log Files?

Active & In-Active Redo Log files

Oracle Database uses only one redo log file at a time to store redo records written from the redo log buffer.

The redo log file that LGWR is actively writing to is called the current redo log file.

What are Redo Log Files?

Active & In-Active Redo Log files

Redo log files that are required for instance recovery are called active redo log files.

Redo log files that are no longer required for instance recovery are called inactive redo log files.

What are Redo Log Files?

Overwriting Redo Log files

If you have enabled archiving (the database is in ARCHIVELOG mode), then the database cannot reuse or overwrite an active online log file until one of the **archiver background processes** (ARCn) has archived its contents.

What are Redo Log Files?

Overwriting Redo Log files

A log switch is the point at which the database stops writing to one redo log file and begins writing to another. Normally, a log switch occurs when the current redo log file is completely filled and writing must continue to the next redo log file.

Oracle Database assigns each redo log file a **new log sequence** number every time a log switch occurs and LGWR begins writing to it.

Introduction to Database Administration

TOPIC 60

Planning Redo Log Files

Planning Redo Log Files

Planning Log Files

Multiplexing Redo Log Files?

Example of Multiplexing

Legal & Illegal Multiplexed Redo Log Configuration

Placing Redo Logs on Different Locations

Adding Redo Log Files to a Group

Planning Redo Log Files

Multiplexing Redo Log Files

To protect the failure of Redo Log Files

Two or more identical copies of Redo Log Files maintained in separate locations.

Files at same location can cause I/O errors, file corruption etc.

Planning Redo Log Files

Multiplexing Redo Log Files

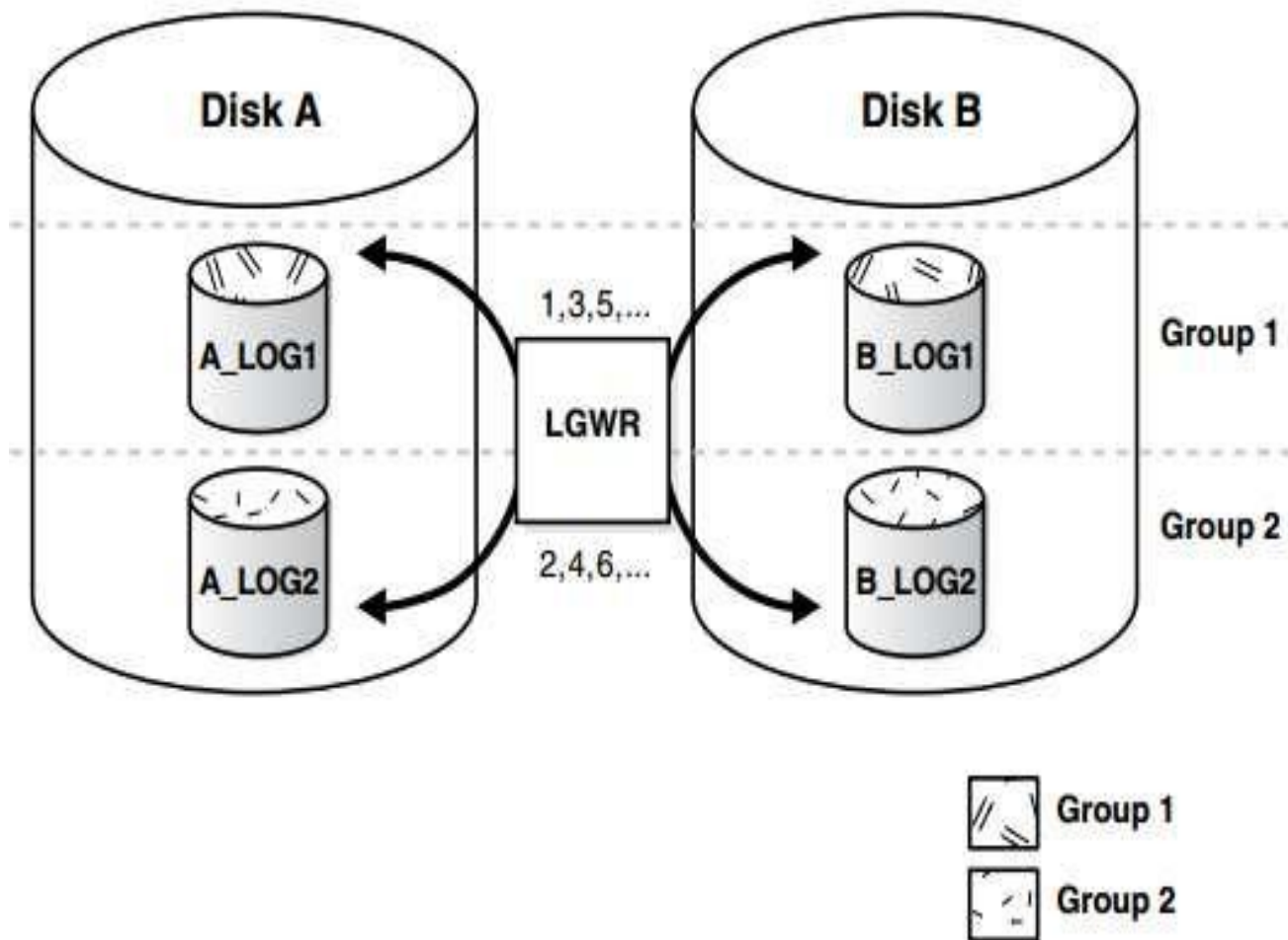
LGWR concurrently writes the same information to multiple identical Redo Log files

A group consists of a Redo Log file and its multiplexed copies.

Each identical copy is said to be a member of the group. Groups can be group 1, group 2 and so on

Planning Redo Log Files

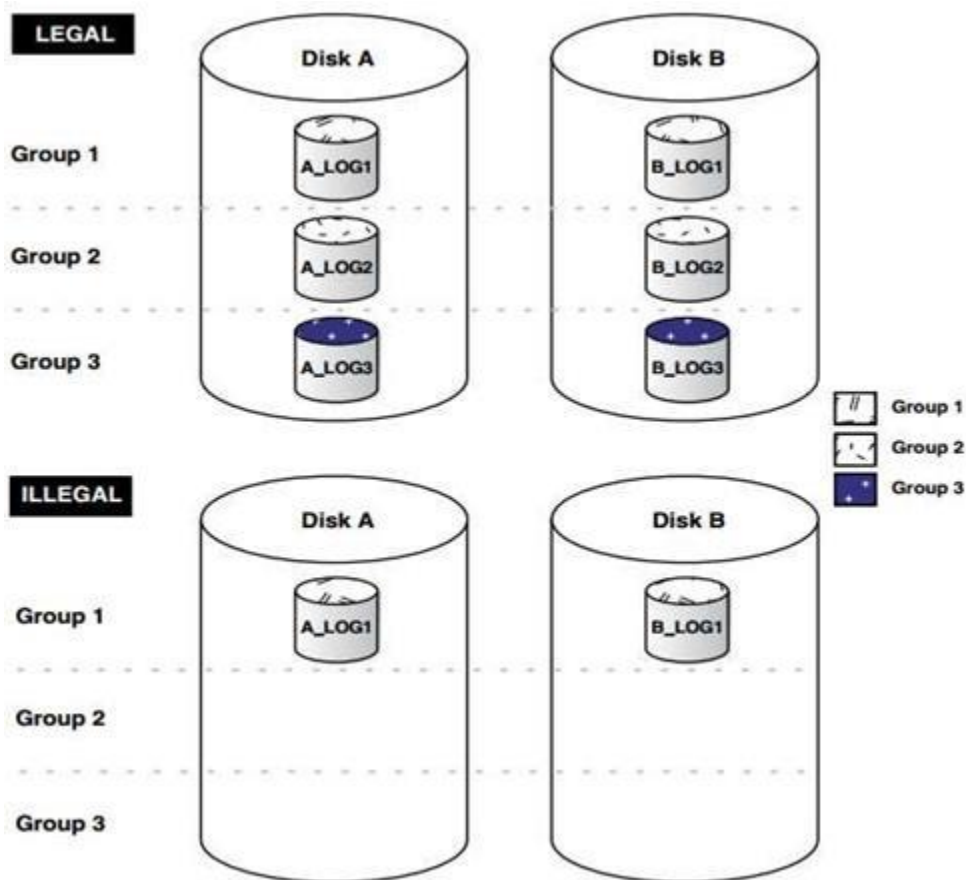
Example of Multiplexing



First LGWR writes concurrently to both A_LOG1 and B_LOG1. Then it writes concurrently to both A_LOG2 and B_LOG2, and so on. LGWR never writes concurrently to members of different groups (for example, to A_LOG1 and B_LOG2)

Planning Redo Log Files

Legal & Illegal Multiplexed Redo Log Configuration



Planning Redo Log Files

Placing Redo Log on Different Locations?

When setting up a multiplexed redo log, place members of a group on different physical disks. If a single disk fails, then only one member of a group becomes unavailable to LGWR and other members remain accessible to LGWR, so the instance can continue to function.

Minimum size permitted to Redo Log File is 4MB.

Planning Redo Log Files

Planning the Block Size of Redo Log Files

```
SQL> SELECT name, value FROM v$sysstat WHERE name = 'redo wastage';
```

NAME	VALUE
redo wastage	4700476

Unlike the database block size, which can be between 2K and 32K, redo log files always default to a block size that is equal to the physical sector size of the disk. Historically, this has typically been 512 bytes (512B).

Some newer high-capacity disk drives offer 4Kbyte (4K) sector sizes for both increased ECC capability and improved format efficiency.

Planning Redo Log Files

Adding Redo Log File to a Group

```
ALTER DATABASE orcl ADD LOGFILE GROUP 4  
('/u01/logs/orcl/redo04a.log', '/u01/logs/orcl/redo04b.log') SIZE 100M BLOCKSIZE 512  
REUSE;
```

```
SQL> select group#, members, BLOCKSIZE FROM V$LOG;
```

GROUP#	MEMBERS	BLOCKSIZE
1	1	512
2	1	512
3	1	512

```
SQL>
```

End

Introduction to Database Administration

TOPIC 61

Creating Redo Log Files Groups

Creating Redo Log Files Groups

Creating Redo Log Files Groups & Members

Creating Redo Log Files Groups

Specify the number with a Group

Creating Redo Log Members

Alternate way of Creating Redo Log Members

Creating Redo Log Files Groups

Creating Redo Log Files Groups

To create a new group of redo log files, use the SQL statement ALTER DATABASE with the ADD LOGFILE clause.

The following statement adds a new group of redo logs to the database:

```
ALTER DATABASE ADD LOGFILE  
('/oracle/dbs/log1c.rdo', '/oracle/dbs/log2c.rdo') SIZE 100M;  
Creating Redo Log Files Groups
```

Specify the number with a Group

To a group number of redo log files, use the SQL statement ALTER DATABASE with the ADD LOGFILE clause.

```
ALTER DATABASE  
ADD LOGFILE GROUP 10  
('/oracle/dbs/log1c.rdo', '/oracle/dbs/log2c.rdo') SIZE 100M BLOCKSIZE 512;  
Creating Redo Log Files Groups
```

Specify the number with a Group

Group number must be between 1 and MAXLOGFILES. Do not skip redo log file group, you will consume unnecessary space in the control files of the database.

Creating Redo Log Files Groups

Creating Redo Log Members

To create new redo log members for an existing group, use the SQL statement ALTER DATABASE with the ADD LOGFILE MEMBER clause. The following statement adds a new redo log member to redo log group number 2:

```
ALTER DATABASE  
ADD LOGFILE MEMBER  
'/oracle/dbs/log2b.rdo' TO GROUP 2;  
Creating Redo Log Files Groups
```

Creating Redo Log Members

Sometimes, some members of groups dropped due to disk failure, you can add new members. Use the following SQL

```
ALTER DATABASE  
ADD LOGFILE MEMBER
```

'/oracle/dbs/log2b.rdo' TO GROUP 2;
Creating Redo Log Files Groups

Alternate way of Creating Redo Log Members

When using the ALTER DATABASE statement, you can alternatively identify the target group by specifying all of the other members of the group in the TO clause, as shown in the following example:

```
ALTER DATABASE ADD LOGFILE MEMBER  
'/oracle/dbs/log2c.rdo'  
TO ('/oracle/dbs/log2a.rdo', '/oracle/dbs/log2b.rdo');
```

END

Introduction to Database Administration

TOPIC 62

Relocating and Renaming Redo Log Members

Relocating and Renaming Redo Log Members

Relocating and Renaming Redo Log Members

Reasons for Relocating Logs

Renaming Redo Log Members

What to do before Relocation?

Steps for Relocating Logs

Metadata of Redo Logs

Relocating and Renaming Redo Log Members

Reasons for Relocating Redo Log

You can use operating system commands to relocate redo logs, then use the ALTER DATABASE statement to make their new names (locations) known to the database.

This procedure is necessary, for example, if the disk currently used for some redo log files is going to be removed, or if data files and several redo log files are stored on the same disk and should be separated to reduce contention.

Relocating and Renaming Redo Log Members

Renaming Redo Log Members

To rename redo log members, you must have the ALTER DATABASE system privilege. Additionally, you might also need operating system privileges to copy files to the desired location and privileges to open and back up the database.

Relocating and Renaming Redo Log Members

What to do before Relocations?

Before relocating your redo logs, or making any other structural changes to the database, completely **BACKUP** the database in case you experience problems while performing the operation. As a precaution, after renaming or relocating a set of redo log files, immediately back up the database control file.

Relocating and Renaming Redo Log Members

Steps for Relocating Redo Logs?

The log files are located on two disks: disk a and disk b.

The redo log is duplexed: one group consists of the members

/diska/logs/log1a.rdo and

/diskb/logs/log1b.rdo, and the second group consists of the members

/diska/logs/log2a.rdo and

/diskb/logs/log2b.rdo.

Relocating and Renaming Redo Log Members

Steps for Relocating Redo Logs?

The redo log files located on disk a must be relocated to disk c. The new filenames will reflect the new location: /diskc/logs/log1c.rdo and

/diskc/logs/log2c.rdo

Relocating and Renaming Redo Log Members

Metadata of Redo Logs?

The redo records have all relevant metadata for the change, including the following:

SCN and time stamp of the change

Transaction ID of the transaction that generated the change

SCN and time stamp when the transaction committed (if it committed)

Type of operation that made the change

Name and type of the modified data segment: SCN: System Change Number

END

Introduction to Database Administration

TOPIC 63

Dropping Redo Log Groups & Members

Dropping Redo Log Groups & Members

Dropping Redo Log Groups & Members

Cases What to Do?

Dropping Redo Log Groups

Dropping Redo Log Members

Dropping Redo Log Groups & Members

Cases What to Do?

In some cases, you may want to drop an entire group of redo log members. For example, you want to reduce the number of groups in an instance redo log.

In a different case, you may want to drop one or more specific redo log members. For example, if a disk failure occurs, you may need to drop all the redo log files on the failed disk so that the database does not try to write to the inaccessible files.

Dropping Redo Log Groups & Members

Cases What to Do?

In other situations, particular redo log files become unnecessary. For example, a file might be stored in an inappropriate location.

Dropping Redo Log Groups & Members

Dropping Redo Log Groups?

To drop a redo log group, you must have the ALTER DATABASE system privilege. Before dropping a redo log group, consider the following restrictions and precautions:

An instance requires at least two groups of redo log files, regardless of the number of members in the groups. (A group comprises one or more members.)

Dropping Redo Log Groups & Members

Dropping Redo Log Groups?

You can drop a redo log group only if it is inactive. If you must drop the current group, then first force a log switch to occur.

Dropping Redo Log Groups & Members

Dropping Redo Log Groups?

Make sure a redo log group is archived (if archiving is enabled) before dropping it. To see whether this has happened, use the V\$LOGview.

Dropping Redo Log Groups & Members

Dropping Redo Log Groups?

```
SELECT GROUP#, ARCHIVED, STATUS FROM V$LOG;  
GROUP# ARC STATUS
```

YES ACTIVE

NO CURRENT

YES INACTIVE

YES INACTIVE

Drop a redo log group with the SQL statement ALTER DATABASE with the DROP LOGFILE clause.

Dropping Redo Log Groups & Members

Dropping Redo Log Groups?

Drop a redo log group with the SQL statement ALTER DATABASE with the DROP LOGFILE clause.

The following statement drops redo log group number 3:

Dropping Redo Log Groups & Members

Dropping Redo Log Groups?

```
ALTER DATABASE DROP LOGFILE GROUP 3;
```

After dropping a redo log group, ensure that the drop completed successfully, and then use the appropriate operating system command to delete the dropped redo log files.

Dropping Redo Log Groups & Members

Dropping Redo Log Members?

Consider the following restrictions and precautions before dropping individual redo log members:

It is permissible to drop redo log files so that a multiplexed redo log becomes temporarily asymmetric. For example, if you use duplexed groups of redo log files, you can drop one member of one group, even though all other groups have two members each.

Dropping Redo Log Groups & Members

Dropping Redo Log Members?

However, you should rectify this situation immediately so that all groups have at least two members, and thereby eliminate the single point of failure possible for the redo log.

Dropping Redo Log Groups & Members

Dropping Redo Log Members?

An instance always requires at least two valid groups of redo log files, regardless of the number of members in the groups. (A group comprises one or more members.) If the member you want to drop is the last valid member of the group, you cannot drop the member until the other members become valid.

To see a redo Forcing Log Switches log file status, use the V\$LOGFILE view.

Dropping Redo Log Groups & Members

Dropping Redo Log Members?

You can drop a redo log member only if it is not part of an active or current group. o drop a member of an active group, first force a log switch to occur.

Make sure the group to which a redo log member belongs is archived (if archiving is enabled) before dropping the member. To see whether this has happened, use the V\$LOG view.

Dropping Redo Log Groups & Members

Dropping Redo Log Members?

To drop specific inactive redo log members, use the following statement:

```
ALTER DATABASE DROP LOGFILE MEMBER  
'/oracle/dbs/log3c.rdo';
```

End

Introduction to Database Administration

TOPIC 64

Forcing Log Switches

Between Redo Log Groups & Members

Forcing Log Switching Between Groups & Members

Forcing Log Switches

Log Switching

Log Switching – An Example

Verifying Blocks in Redo Log Files

Forcing Log Switching Between Groups & Members

Log Switching

A log switch occurs when LGWR stops writing to one redo log group and starts writing to another. By default, a log switch occurs automatically when the current redo log file group fills. You can force a log switch to make the currently active group inactive and available for redo log maintenance operations.

For example, you want to drop the currently active group, but are not able to do so until the group is inactive.

Forcing Log Switching Between Groups & Members

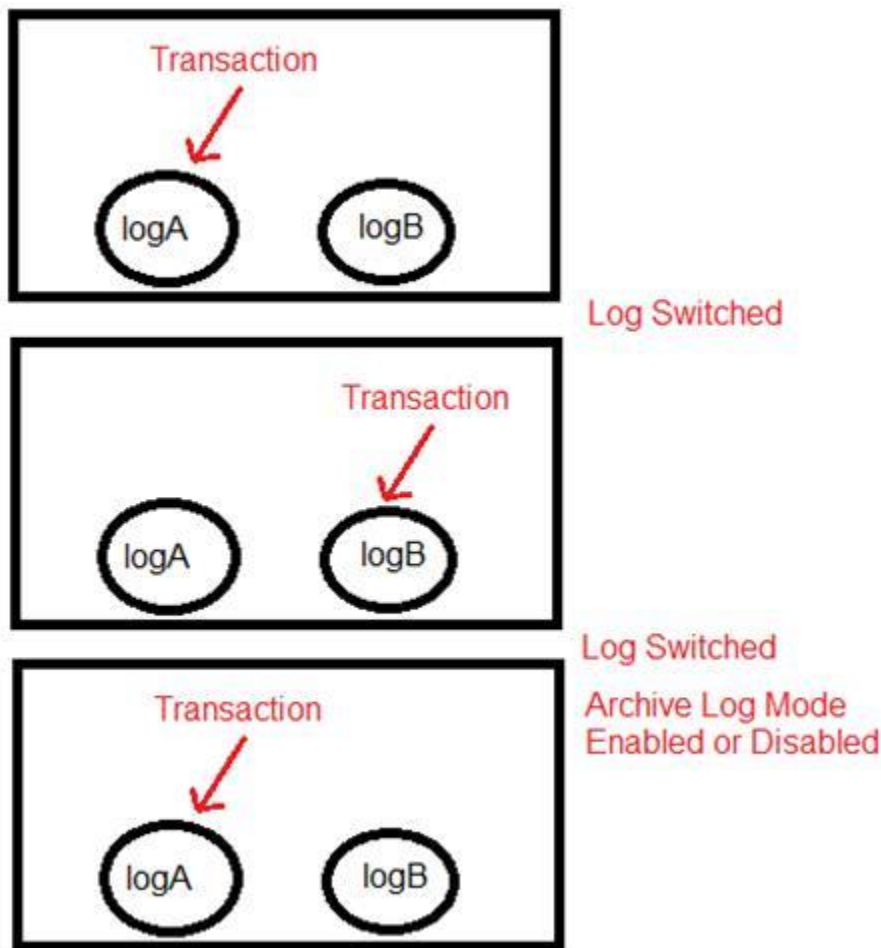
Log Switching

You may also want to force a log switch if the currently active group must be **archived** at a specific time before the members of the group are completely filled. This option is useful in configurations with large redo log files that take a long time to fill.

The following statement forces a log switch:

```
ALTER SYSTEM SWITCH LOGFILE;
```

Forcing Log Switching Between Groups & Members



Log Switching – An Example

Forcing Log Switching Between Groups & Members

Verifying Blocks in Redo Log Files

You can configure the database to use checksums to verify blocks in the redo log files.

If you set the initialization parameter `DB_BLOCK_CHECKSUM` to `TYPICAL`(the default), the database computes a check sum for each database block when it is written to disk, including each redo log block as it is being written to the current log.

The checksum is stored header's block.

Forcing Log Switching Between Groups & Members

Verifying Blocks in Redo Log Files

Oracle Database uses the checksum to detect corruption in a redo log block.

The database verifies the redo log block when the block is read from an archived log during recovery and when it writes the block to an archive log file. An error is raised and written to the alert log if corruption is detected.

Forcing Log Switching Between Groups & Members

Verifying Blocks in Redo Log Files

If corruption is detected in a redo log block while trying to archive it, the system attempts to read the block from another member in the group.

If the block is corrupted in all members of the redo log group, then archiving cannot proceed.

The value of the DB_BLOCK_CHECKSUM parameter can be changed dynamically using the ALTER SYSTEM statement.

END

Introduction to Database Administration

TOPIC 65

Clearing Redo Log File

Clearing Redo Log File

Clearing Redo Log Files

How to clear Redo Log Files?

Error in clearing Redo Log Files

Dropping Redo Log File

Creating Redo Log File

Clearing Redo Log File

How to clear Redo Log Files?

A redo log file might become corrupted while the database is open, and ultimately stop database activity because archiving cannot continue. In this situation the ALTER DATABASE CLEAR LOGFILE statement can be used to reinitialize the file without shutting down the database.

The following statement clears the log files in redo log group number 3:

```
ALTER DATABASE CLEAR LOGFILE GROUP 3;
```

Clearing Redo Log File

How to clear Redo Log Files?

If you clear a log file that is needed for recovery of a backup, then you can no longer recover from that backup. The database writes a message in the alert log describing the backups from which you cannot recover.

To clear an unarchived redo log that is needed to bring an offline tablespace online, use the UNRECOVERABLE DATAFILE clause in the ALTER DATABASE CLEAR LOGFILE statement.

Clearing Redo Log File

Error in clearing Redo Log Files

```
SQL> alter database drop logfile group 1; alter database drop logfile group 1
*
```

ERROR at line 1:

ORA-01624: log 1 needed for crash recovery of instance SBIP18DB (thread 1)

ORA-00312: online log 1 thread 1: '/SIBIP16/SBIP18DB/SBIP18DB/redo01.log'

Clearing Redo Log File

Error in clearing Redo Log Files

```
select l.group#, l.thread#, substr(f.member,1,50) f_name, l.archived,
l.status, (bytes/1024/1024) fsize from
v$log l, v$logfile f
where f.group# = l.group#
```

D#	F_NAME	ARC	STATUS	FSIZE
1	E:\APP\HAIDER\ORADATA\ORCL\REDO01.LOG	NO	INACTIVE	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO02.LOG	NO	INACTIVE	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO03.LOG	NO	CURRENT	50

order by 1,2;

Clearing Redo Log File

Error in clearing Redo Log Files

We can see the though the redolog group 1 has been archived, its status is ACTIVE.

i.e this redolog is required for crash recovery or block recovery.

So to change the status from ACTIVE to INACTIVE, switch logfile multiple times, till the status is INACTIVE.

#	F_NAME	ARC	STATUS	FSIZE
1	E:\APP\HAIDER\ORADATA\ORCL\REDO01.LOG	NO	CURRENT	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO02.LOG	NO	INACTIVE	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO03.LOG	NO	ACTIVE	50

```
alter system switch logfile;
--for inactive status
alter database drop logfile group 2;
Clearing Redo Log File
```

Error in clearing Redo Log Files?

We can see the though the redolog group 1 has been archived, its status is ACTIVE.
i.e this redolog is required for crash recovery or block recovery.

D#	F_NAME	ARC	STATUS	FSIZE
1	E:\APP\HAIDER\ORADATA\ORCL\REDO01.LOG	NO	ACTIVE	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO02.LOG	NO	CURRENT	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO03.LOG	NO	INACTIVE	50

So to change the status from ACTIVE to INACTIVE, switch logfile multiple times, till the status is INACTIVE.

```
alter system switch logfile;
--for inactive status
alter database drop logfile group 3;
Clearing Redo Log File
```

Dropping Redo Log File

```
SQL> alter database drop logfile group 3;
Database altered.

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area 1071333376 bytes
Fixed Size                  1375792 bytes
Variable Size               713032144 bytes
Database Buffers            352321536 bytes
Redo Buffers                 4603904 bytes
Database mounted.
Database opened.
SQL>
```

You shutdown and startup database after dropping on redo group 3. Drop REDO03.LOG from folder.

Clearing Redo Log File

Creating Redo Log File

alter database add logfile group 3 'E:\app\Haider\oradata\orcl\REDO03.log' size 50m;

D#	F_NAME	ARC	STATUS	FSIZE
1	E:\APP\HAIDER\ORADATA\ORCL\REDO01.LOG	NO	INACTIVE	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO02.LOG	NO	CURRENT	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO03.LOG	YES	UNUSED	50

Introduction to Database Administration

TOPIC 66

Metadata Views of Redo Log File

Metadata Views of Redo Log File

Metadata View of Redo Log Files

Instance & Database Log Buffer

Significant Views?

Control File Information

Log History Information

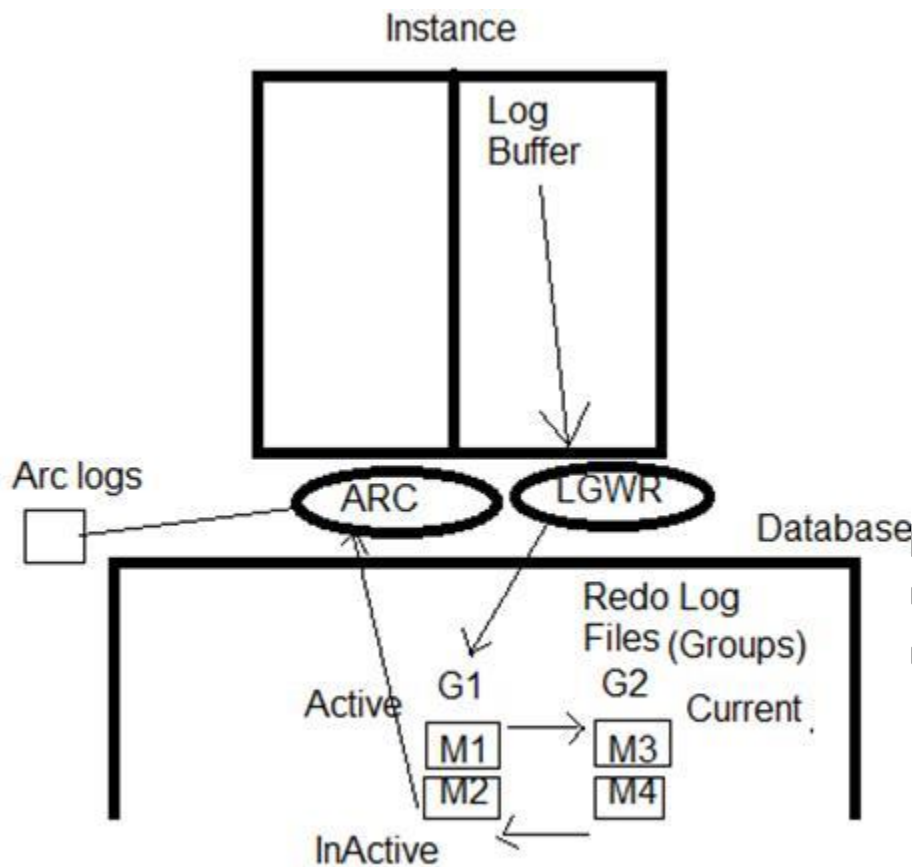
Log of Members of Group

Log History Information

Use of Members of Group with Status

Metadata Views of Redo Log File

Instance & Database Log Buffer



Metadata Views of Redo Log File

Significant of Views

View	Description
V\$LOG	Displays the redo log file information from the control file
V\$LOGFILE	Identifies redo log groups and members and member status
V\$LOG_HISTORY	Contains log history information

Metadata Views of Redo Log File

Control File Information

```
SELECT * FROM V$LOG;
```

GROUP#	THREAD#	SEQ	BYTES	MEMBERS	ARC	STATUS	FIRST_CHANGE#	FIRST_TIM
1	1	10605	1048576	1	YES	ACTIVE	11515628	16-APR-00
2	1	10606	1048576	1	NO	CURRENT	11517595	16-APR-00
3	1	10603	1048576	1	YES	INACTIVE	11511666	16-APR-00
4	1	10604	1048576	1	YES	INACTIVE	11513647	16-APR-00

Metadata Views of Redo Log File Members of Group

```
SELECT * FROM V$LOG;
```

GROUP#	THREAD#	SEQ	BYTES	MEMBERS	ARC	STATUS	FIRST_CHANGE#	FIRST_TIM
1	1	10605	1048576	1	YES	ACTIVE	11515628	16-APR-00
2	1	10606	1048576	1	NO	CURRENT	11517595	16-APR-00
3	1	10603	1048576	1	YES	INACTIVE	11511666	16-APR-00
4	1	10604	1048576	1	YES	INACTIVE	11513647	16-APR-00

Metadata Views of Redo Log File Log History Information

```
SQL> select Stamp, Thread#, First_change#, First_time, Resetlogs_change#
2 from v$log_history
3 where rownum<=10;
```

STAMP	THREAD#	FIRST_CHANGE#	FIRST_TIM	RESETLOGS_CHANGE#
951399757	1	9079208	07-AUG-17	940976
951473702	1	9104778	07-AUG-17	940976
951559221	1	9132765	08-AUG-17	940976
951645623	1	9157391	09-AUG-17	940976
951732023	1	9182645	10-AUG-17	940976
952077284	1	9204703	11-AUG-17	940976
952085754	1	9217064	15-AUG-17	940976
952164052	1	9247525	15-AUG-17	940976
952303136	1	9278762	16-AUG-17	940976
952339592	1	9307368	18-AUG-17	940976

Metadata Views of Redo Log File

Use of member with status

```
select l.group#, l.thread#, substr(f.member,1,50) f_name, l.archived,  
l.status, (bytes/1024/1024) fsize from  
v$log l, v$logfile f
```

D#	F_NAME	ARC	STATUS	FSIZE
1	E:\APP\HAIDER\ORADATA\ORCL\REDO01.LOG	NO	INACTIVE	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO02.LOG	NO	INACTIVE	50
1	E:\APP\HAIDER\ORADATA\ORCL\REDO03.LOG	NO	CURRENT	50

where f.group# = l.group# order by 1,2;

Introduction to Database Administration

TOPIC 67

Archive & NoArchive Redo Log Files

Archive and

Archive & NoArchive of Redo Log Files

Archived Redo Log

Archived Redo Log Usage

Archived Redo Log Mode

Archived Redo Log Disabled to Enabled

Archived Redo Log Enabled to Disabled

Redo Logs Use in ARCHIVELOG Mode

Archive and NoArchive Redo Log Files

Archived Redo Log

Oracle Database lets you save filled groups of redo log files to one or more offline destinations, known collectively as the archived redo log.

The process of turning redo log files into archived redo log files is called archiving. This process is only possible if the database is running in ARCHIVELOG mode.

You can choose automatic or manual archiving
Archive and Redo Log Files

Archived Redo Log

An archived redo log file is a copy of one of the filled members of a redo log group with log sequence number.

For example, if you are multiplexing your redo log, and if group 1 contains identical member files a_log1 and b_log1, then the archiver process (ARCn) will archive one of these member files.

Archive Redo Log Files

Archived Redo Log

Should a_log1 become corrupted, then ARCn can still archive the identical b_log1. The archived redo log contains a copy of every group created since you enabled archiving.

Archive and Log Files

Archived Redo Log

When the database is running in ARCHIVELOG mode, LGWR cannot reuse and hence overwrite a redo log group until it has been archived.

The background process RCn automates archiving operations when automatic archiving is enabled. The database starts multiple archiver processes as needed to ensure that the archiving of filled redo logs does not fall behind.

Archive and

Archived Redo Log Usage

You can use archived redo logs to:

- Recover a database

- Update a standby database

- Get information about the history of a database using the LogMiner utility

Archive Log Files

Archived Redo Log Mode

Display DB Archive Mode Disable or Enable Login with sys / as sysdba.

We will discuss archival files backup later

```
SQL> SHOW PARAMETER DB_RECOVERY_FILE_DEST
```

NAME	TYPE	VALUE
db_recovery_file_dest	string	E:\app\Haider\flash_recovery_area
db_recovery_file_dest_size	big integer	3852M

```
SQL>
```

Archive and NoArchive Redo Log Files

```
SQL> shutdown immediate
ORA-01507: database not mounted

ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area 1071333376 bytes
Fixed Size 1375792 bytes
Variable Size 713032144 bytes
Database Buffers 352321536 bytes
Redo Buffers 4603904 bytes
Database mounted.
SQL> Alter database archivelog;

Database altered.

SQL> archive log list
Database log mode Archive Mode
Automatic archival Enabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 760
Next log sequence to archive 762
Current log sequence 762
```

```
SQL> shutdown immediate
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
SQL> startup open
ORACLE instance started.

Total System Global Area 1071333376 bytes
Fixed Size 1375792 bytes
Variable Size 713032144 bytes
Database Buffers 352321536 bytes
Redo Buffers 4603904 bytes
Database mounted.
Database opened.
```

Archived Redo Log Disabled to Enabled

Using SQL sys to convert Archivelog Disabled to Enabled & Vice Versa

During recovery Oracle ask you to use
Alter database open instead of startup open
Archive and NoArchive Redo Log Files

Archived Redo Log Enabled to Disabled

```

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area 1071333376 bytes
Fixed Size                  1375792 bytes
Variable Size               713032144 bytes
Database Buffers            352321536 bytes
Redo Buffers                 4603904 bytes
Database mounted.
SQL> alter database noarchivelog;

Database altered.

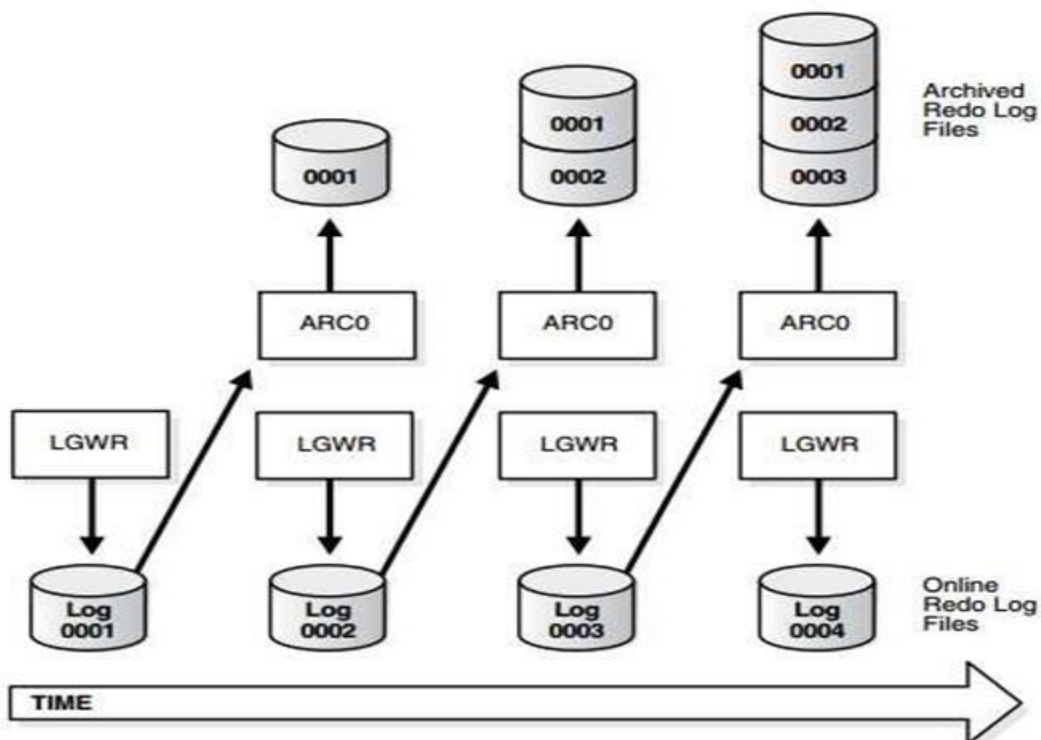
SQL> archive log list
Database log mode              No Archive Mode
Automatic archival            Disabled
Archive destination           USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence    760
Current log sequence          762
SQL> startup open
ORA-01081: cannot start already-running ORACLE - shut it down first
SQL> shutdown immediate
ORA-01109: database not open

Database dismounted.
ORACLE instance shut down.
SQL> startup open
ORACLE instance started.

```

Archive and NoArchive Redo Log Files

Redo Logs Use in ARCHIVELOG Mode



TOPIC 68

Controlling Archiving

Controlling Archiving

Controlling Archive of Redo Log

What is Oracle Archiving?

Setting the Initial Database Archiving Mode

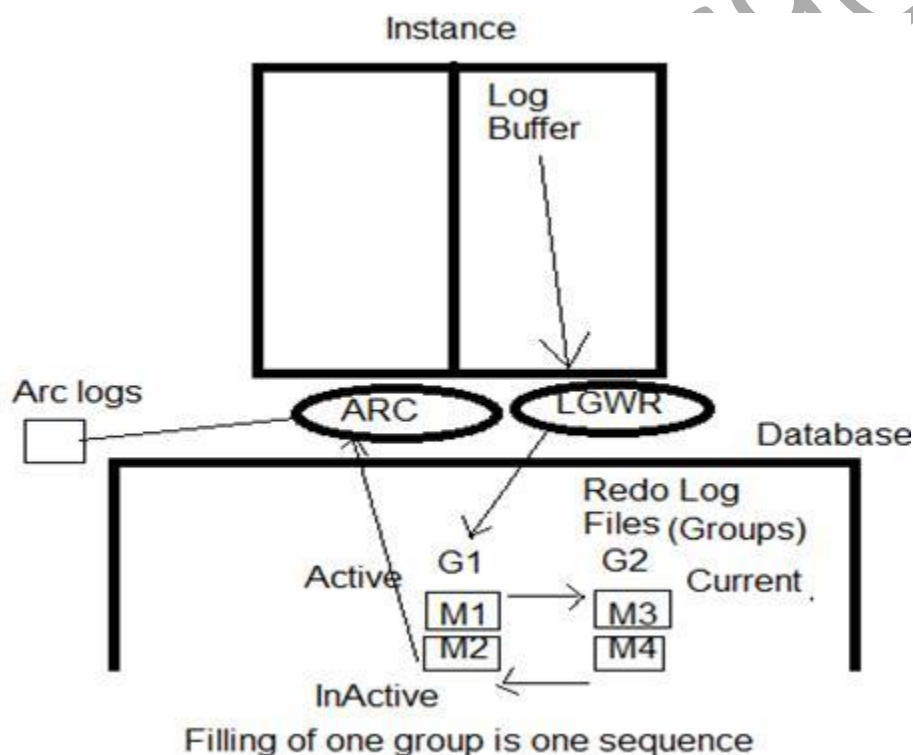
Changing the Database Archiving Mode

Performing Manual Archiving

Adjusting the Number of Archiver Processes

Controlling Archiving

What is Oracle Archiving?

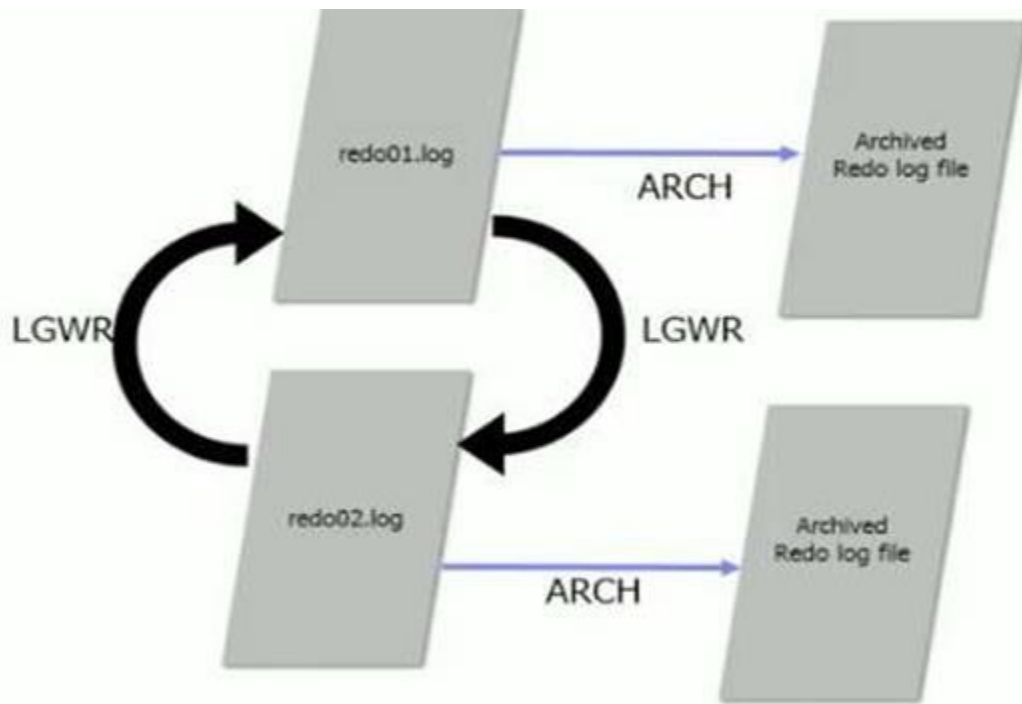


Archiving: After a switch, a copy of the Redo Log file is sent to Archive Destination

Controlling Archiving

What is Oracle Archiving?

Archiving: After a switch, a copy of the Redo Log file is sent to Archive Destination



Controlling Archiving

Setting the Initial Database Archiving Mode

Usually, you can use the default of NOARCHIVELOG mode at database creation because there is no need to archive the redo information generated by that process.

After creating the database, decide whether to change the initial archiving mode.

Controlling Archiving

Setting the Initial Database Archiving Mode

If you specify ARCHIVELOG mode, you must have initialization parameters set that specify the destinations for the archived redo log files

Controlling Archiving

Changing Archiving Mode

Start a new instance and mount, but do not open, the database.

STARTUP MOUNT

To enable or disable archiving, the database must be mounted but not open.

Controlling Archiving

Changing Archiving Mode

Change the database archiving mode. Then open the database for normal operations.

```
ALTER DATABASE ARCHIVELOG; ALTER DATABASE OPEN;
```

Shut down the database. SHUTDOWN IMMEDIATE

Back up the database.

Controlling Archiving

Changing Archiving Mode Manual

Start a new instance and mount, but do not open, the database.

```
STARTUP MOUNT
```

Change the database archiving mode. Then open the database for normal operations.

```
ALTER DATABASE ARCHIVELOG MANUAL; ALTER DATABASE OPEN;
```

```
ALTER SYSTEM ARCHIVELOG ALL;
```

Shut down the database.

```
SHUTDOWN IMMEDIATE
```

Back up the database.

Controlling Archiving

Adjusting the Number of Archiver Processes

The LOG_ARCHIVE_MAX_PROCESSES initialization parameter specifies the number of ARCn processes that the database initially starts. The default is four processes.

To avoid any run-time overhead of starting additional ARCn processes, you can set the

```
LOG_ARCHIVE_MAX_PROCESSES
```

Controlling Archiving

Adjusting the Number of Archiver Processes

Initialization parameter to specify that up to 30 ARCn

```
ALTER SYSTEM SET
```

```
LOG_ARCHIVE_MAX_PROCESSES=6;
```

End

Introduction to Database Administration

TOPIC 69

Specifying Archive

Destinations

Specifying Archive Destinations

Specifying Archive Destinations

Specifying Archive Destinations

Archive Destinations Status

Specifying Archive Destinations

Specifying Archive Destinations

You can choose to archive redo logs to a single destination or to multiple destinations. Destinations can be local—within the local file system or an Oracle Automatic Storage Management (Oracle ASM) disk group—or remote.

Specifying Archive Destinations

Specifying Archive Destinations

When you archive to multiple destinations, a copy of each filled redo log file is written to each destination. These redundant copies help ensure that archived logs are always available in the event of a failure at one of the destinations.

Specifying Archive Destinations

Specifying Archive Destinations

To archive to only a single destination, specify that destination using the LOG_ARCHIVE_DEST initialization parameter.

To archive to multiple destinations, you can choose to archive to two or more locations using the

Specifying Archive Destinations

Specifying Archive Destinations

LOG_ARCHIVE_DEST_n initialization parameters, or to archive only to a primary and secondary destination using the LOG_ARCHIVE_DEST and

LOG_ARCHIVE_DUPLEX_DEST initialization parameters.

Specifying Archive Destinations

Specifying Archive Destinations

Method	Initialization Parameter	Host	Example
1	LOG_ARCHIVE_DEST_ <i>n</i> where: <i>n</i> is an integer from 1 to 31. Archive destinations 1 to 10 are available for local or remote locations. Archive destinations 11 to 31 are available for remote locations only.	Local or remote	LOG_ARCHIVE_DEST_1 = 'LOCATION=/disk1/arc' LOG_ARCHIVE_DEST_2 = 'LOCATION=/disk2/arc' LOG_ARCHIVE_DEST_3 = 'SERVICE=standby1'
2	LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST	Local only	LOG_ARCHIVE_DEST = '/disk1/arc' LOG_ARCHIVE_DUPLEX_DEST = '/disk2/arc'

Specifying Archive Destinations

Archive Destinations Status

Valid/Invalid: indicates whether the disk location or service name information is specified and valid

Enabled/Disabled: indicates the availability state of the location and whether the database can use the destination

Active/Inactive: indicates whether there was a problem accessing the destination

Introduction to Database Administration

TOPIC 70

Managing Archive Destination Failure & Controlling Trace

Managing Archive Destination Failure & Controlling Trace

Managing Archive Destination Failure & Controlling Trace

Rearchiving to a Failed Destination

Controlling Trace Output

Initialization Parameter

Archived Redo Log Views

Managing Archive Destination Failure & Controlling Trace

Rearchiving to a Failed Destination

Use the REOPEN attribute of the LOG_ARCHIVE_DEST_n parameter to specify whether and when ARCn should attempt to rearchive to a failed destination following an error. REOPEN applies to all errors, not just OPEN errors.

Managing Archive Destination Failure & Controlling Trace

Rearchiving to a Failed Destination

REOPEN=n sets the minimum number of seconds before ARCn should try to reopen a failed destination. The default value for n is 300 seconds. A value of 0 is the same as turning off the REOPEN attribute; ARCn will not attempt to archive after a failure.

If you do not specify the REOPEN keyword, ARCn will never reopen a destination following an error.

Managing Archive Destination Failure & Controlling Trace

Rearchiving to a Failed Destination

If you do not specify the REOPEN keyword, ARCn will never reopen a destination following an error.

You cannot use REOPEN to specify the number of attempts ARCn should make to reconnect and transfer archived logs. The REOPEN attempt either succeeds or fails.

You cannot use REOPEN to specify the number of attempts ARCn should make to reconnect and transfer archived logs. The REOPEN attempt

Managing Archive Destination Failure & Controlling Trace

Rearchiving to a Failed Destination

When you specify REOPEN for an OPTIONAL destination, the database can overwrite online logs if there is an error. If you specify REOPEN for a MANDATORY destination, the database stalls the production database when it cannot successfully archive.

Managing Archive Destination Failure & Controlling Trace

Rearchiving to a Failed Destination

In this situation, consider the following options:

- Archive manually to the failed destination.

- Change the destination by deferring the destination, specifying the destination as optional, or changing the service.

- Drop the destination.

Managing Archive Destination Failure & Controlling Trace

Controlling Trace Output

Background processes always write to a trace file when appropriate. In the case of the archive log process, you can control the output that is generated to the trace file. You do this by setting the LOG_ARCHIVE_TRACE initialization parameter to specify a trace level.

The following values can be specified on next slide:

Managing Archive Destination Failure & Controlling Trace

Initialization Parameter

Trace Level	Meaning
0	Disable archivelog tracing. This is the default.
1	Track archival of redo log file.
2	Track archival status for each archivelog destination.
4	Track archival operational phase.
8	Track archivelog destination activity.
16	Track detailed archivelog destination activity.
32	Track archivelog destination parameter modifications.
64	Track ARC <i>n</i> process state activity.
128	Track FAL (fetch archived log) server related activities.
256	Supported in a future release.
512	Tracks asynchronous LGWR activity.
1024	RFS physical client tracking.
2048	ARC <i>n</i> /RFS heartbeat tracking.
4096	Track real-time apply
8192	Track redo apply activity (media recovery or physical standby)

Managing Archive Destination Failure & Controlling Trace

Archived Redo Log Views

Dynamic Performance View	Description
V\$DATABASE	Shows if the database is in ARCHIVELOG or NOARCHIVELOG mode and if MANUAL (archiving mode) has been specified.
V\$ARCHIVED_LOG	Displays historical archived log information from the control file. If you use a recovery catalog, the RC_ARCHIVED_LOG view contains similar information.
V\$ARCHIVE_DEST	Describes the current instance, all archive destinations, and the current value, mode, and status of these destinations.
V\$ARCHIVE_PROCESSES	Displays information about the state of the various archive processes for an instance.
V\$BACKUP_REDOLOG	Contains information about any backups of archived logs. If you use a recovery catalog, the RC_BACKUP_REDOLOG contains similar information.
V\$LOG	Displays all redo log groups for the database and indicates which need to be archived.
V\$LOG_HISTORY	Contains log history information such as which logs have been archived and the SCN range for each archived log.

```
SELECT GROUP#, ARCHIVED
FROM SYS.V$LOG;
```

```
GROUP#    ARC
-----    --
       1    YES
       2    NO
```

End
Introduction to Database Administration

TOPIC 71

Database Physical Design
Database Physical Design

Physical Design

Definition & Concepts

Criteria for Physical Design

Analyzing DB Queries & Transactions

Time Constraints of Queries & Transactions

Physical DB Design Decisions

Physical Database Design for Tuning

Analyzing Database Queries & Transactions

Database Physical Design

Definition & Concepts

Physical database design is the process of choosing specific storage structures and access paths for the database files to achieve good performance for the various database applications.

Database Physical Design

Definition & Concepts

Each DBMS offers a variety of options for file organization and access path. Each DBMS includes various types of indexing, clustering of related records on disk blocks, link related records via pointers and various types of hashing/ partitions.

Database Physical Design

Definition & Concepts

For example, Oracle gives the concept of partitions as TableSpaces & MS SQL Server provides concept of sub databases

Database Physical Design

Criteria for Physical Design

Response Time: Elapse time between submitting a database transaction for execution and receiving a response. Response time is also influenced by factors not under control of DBMS, factors are communication, scheduling and operating system etc.

Database Physical Design

Criteria for Physical Design

Space Utilization: An amount of storage space used by the database files and their access path structure on disk including indexes.

Transaction Throughput: Average number of transactions processed per minute. It is measured at peak conditions on the system

Database Physical Design

Analyzing DB Queries & Transactions

The files that will be accessed by the query.

Whether the selection condition is an equality, inequality, or a range condition.

The attributes whose values will be retrieved by the query.

The attributes on which any selection conditions for the queries are specified.

Database Physical Design

Analyzing DB Queries & Transactions

The attributes on which any join conditions or conditions to link multiple tables or objects for the query are specified.

The files that will be modified.

The attributes on which selection conditions for a delete or update is specified.

Database Physical Design

Time Constraints of Queries & Transactions

Some queries and transactions may stringent performance constraints.

For example, a transaction may have the constraint that it should terminate within 5 seconds on 95 percent of the occasions when it is invoked than it would never take more than 20 seconds.

The type of operations on each file (insert, delete and update)

Database Physical Design

Physical DB Design Decisions

Design decisions about indexing: Attributes that are joint on equality, inequality or range conditions.

Whether to index an attribute.

What attribute(s) to index on. (Composite keys or one of attributes in key and other is non-key)

Database Physical Design

Physical DB Design Decisions

Whether to set up a Clustered index.

Whether to use hash index over tree index.

Whether to use dynamic hashing for the file (files are very volatiles).

Apply de-normalization: separate the logically related attributes into tables to minimize redundancy.

Database Physical Design

Physical Database Design for Tuning

Size of individual tables

Number of distinct values in a column

Time of query submission and its execution

Storage statistics: Partitions, index space and buffer pool

I/O & performance statistics: Total read/write activity

Database Physical Design

Physical Database Design for Tuning

Locking/ logging related statistics: Rates of issuing different types of locks, transaction throughput rates, and log records activity

Index statistics: Number of levels in an index, number of non-contiguous leaf pages.

Database Physical Design

Analyzing Database Queries & Transactions

The files that will be accessed by the query

The attributes on which any selection condition (WHERE) for the query are specified

The attributes on which any join conditions are specified

Database Physical Design

Analyzing Database Queries & Transactions

The attributes whose values will be retrieved by the queries

The files that will be updated

Type of operations on files (INSERT, DELETE, UPDATE)

The attributes whose values will be changed by an update operation

Introduction to Database Administration

TOPIC 72

Defining Tablespace, Types

& Data Files

Defining

Tablespace & Data Files

Definition & Concepts

Benefits of using Multiple Tablespaces

Assigning Tablespace Quotas to Users

Types of Tablespaces

Defining & Data Files

Definition & Concepts

An Oracle database consists of one or more logical storage units called tablespaces, which collectively store all of the database's data.



For example, Enterprise's data tablespace, temporary tablespace, index tablespace etc.

Defining & Data Files

Definition & Concepts

Tablespaces (Logical Level)

Can belong to only one database at a time

Consists of one or more data files

Are further divided into logical units of storage

Data Files (Physical Level)

Can belong to only one tablespace and one database

Are a repository for schema object data

& Data Files

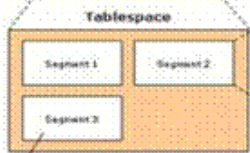
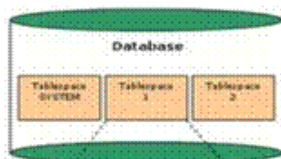
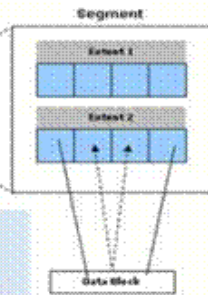
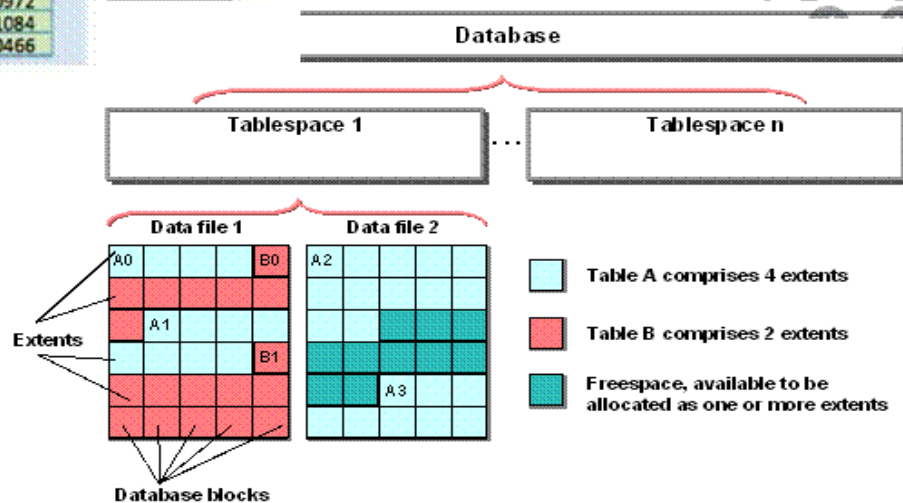


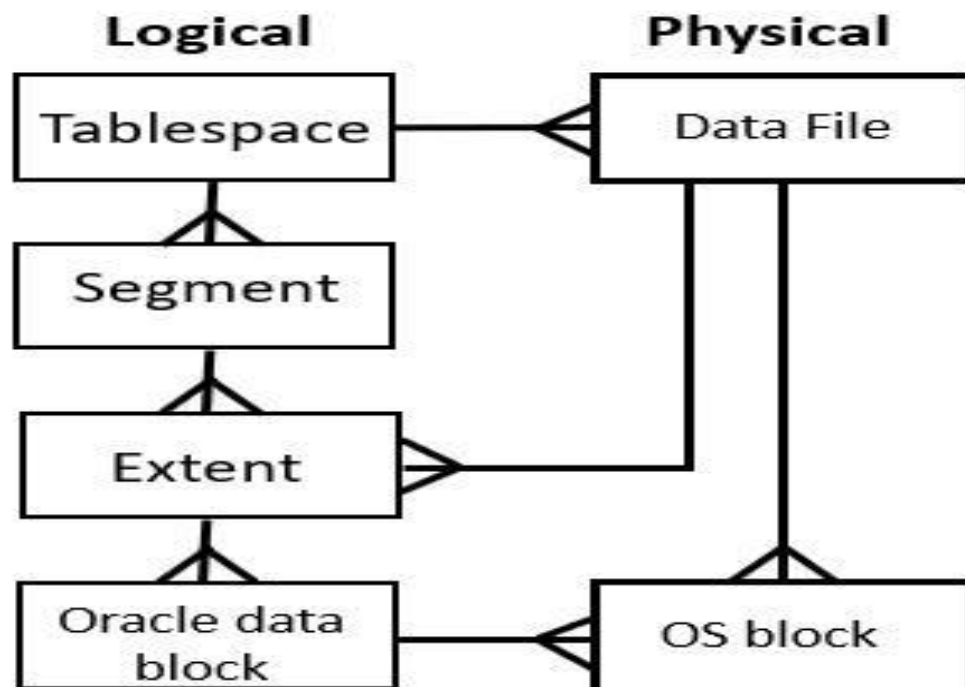
Table	Column
Row	
001	Smith 080972
002	Jones 041084
003	Davis 070466

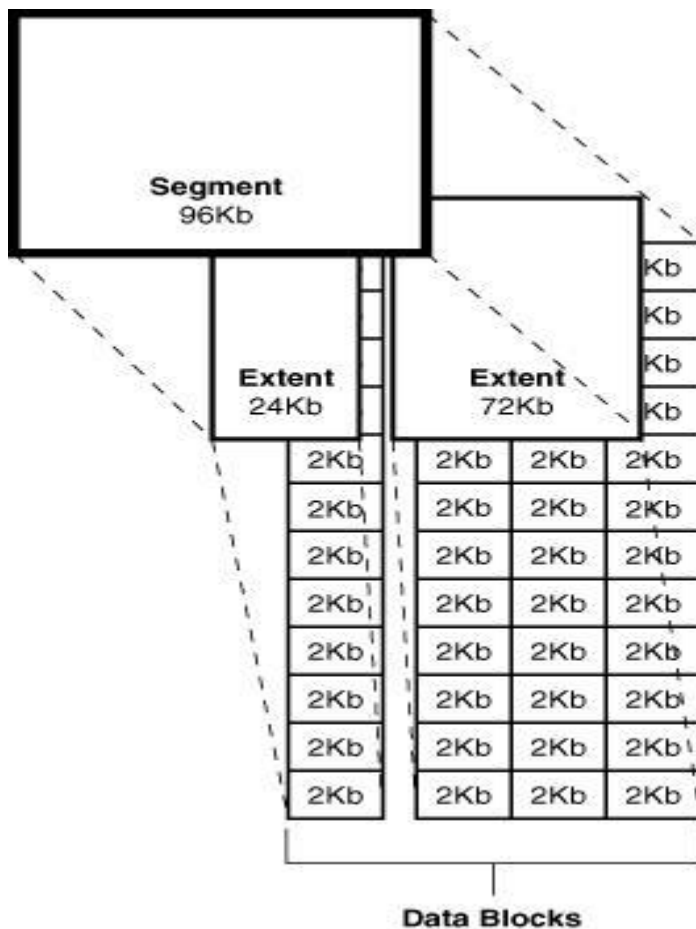


An extent is a logical unit of database storage space allocation made up of a number of contiguous data blocks. One or more extents in turn make up a segment. When the existing space in a segment is completely used, Oracle allocates a new extent for the segment.



Tablespaces & Extents





Defining & Data Files

Benefits of using Multiple Tablespaces

Separate user data from data dictionary data to reduce I/O contention.

Separate data of one application from the data of another to prevent multiple applications for offline purpose.

Store the data files of different tablespaces on different disk drives to reduce I/O contention.

Take individual tablespaces offline while others remain online, providing better overall availability.

& Data Files

Benefits of using Multiple Tablespaces

Optimizing tablespace use by reserving a tablespace for a particular type of database use, such as high update activity, read-only activity, or temporary segment storage.

Back up individual tablespaces.

Note: Some operating system can affect the number of tablespaces that can simultaneously online.

Defining Tablespace, Types & Data Files

Assigning Tablespace Quotas to Users

Grant to users who will be creating tables, clusters, materialized views, indexes, and other objects the privilege to create the object and a quota(space allowance or limit) in the tablespace intended to hold the object segment.

Defining Tablespace, Types & Data Files

Assigning Tablespace Quotas to Users

For PL/SQL objects such as packages, procedures, and functions, users only need the privileges to create the objects. No explicit tablespace quota is required to create these PL/SQL objects

Defining Tablespace, Types & Data Files

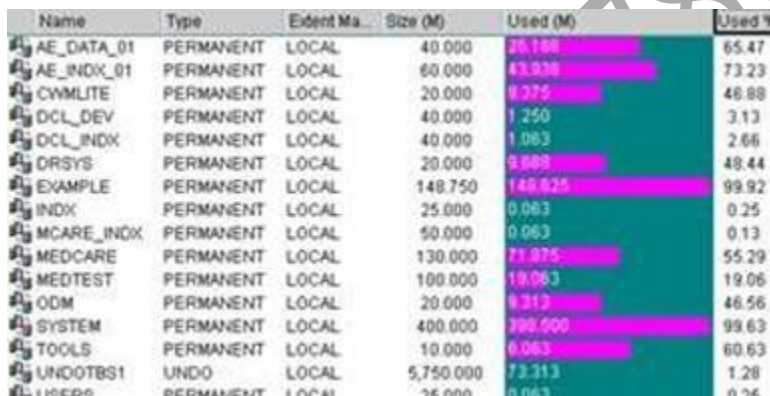
Types of Tablespaces

SYSTEM tablespace

Created with the database

Contains the data dictionary

Contains the SYSTEM undo segment



Name	Type	Extent Ma.	Size (M)	Used (M)	Used %
AE_DATA_01	PERMANENT	LOCAL	40,000	26,188	65.47
AE_INDX_01	PERMANENT	LOCAL	60,000	43,938	73.23
CYMLITE	PERMANENT	LOCAL	20,000	9,375	46.88
DCL_DEV	PERMANENT	LOCAL	40,000	1,250	3.13
DCL_INDX	PERMANENT	LOCAL	40,000	1,063	2.66
DRSYS	PERMANENT	LOCAL	20,000	9,809	49.04
EXAMPLE	PERMANENT	LOCAL	148,750	148,625	99.92
INDX	PERMANENT	LOCAL	25,000	0,063	0.25
MCARE_INDX	PERMANENT	LOCAL	50,000	0,063	0.13
MEDCARE	PERMANENT	LOCAL	130,000	71,875	55.29
MEDTEST	PERMANENT	LOCAL	100,000	19,063	19.06
ODM	PERMANENT	LOCAL	20,000	9,313	46.56
SYSTEM	PERMANENT	LOCAL	400,000	398,000	99.50
TOOLS	PERMANENT	LOCAL	10,000	6,063	60.63
UNDOTBS1	UNDO	LOCAL	5,750,000	73,313	1.28
UNDOTBS2	UNDO	LOCAL	14,000	0,063	0.44

Non-SYSTEM tablespace

Separate segments

Eases space administration

Controls amount of space of a user

Introduction to Database Administration

TOPIC 73

Creating Tablespaces

Creating Tablespaces

Creating Tablespaces

Creating Tablespace Syntax

Creating Tablespace using SQL

How a user uses Tablespace?

Space Management in Tablespace

Locally-Managed Tablespaces

Dictionary-Managed Tablespaces

Create UNDO Tablespace

Create Temporary Tablespace

Creating a default Temporary Tablespace

Restrictions

Creating Tablespaces

Creating Tablespace Syntax

```
CREATE TABLESPACE tablespace
    [DATAFILE clause]
    [MINIMUM EXTENT integer[K|M]]
    [BLOCKSIZE integer [K]]
    [LOGGING|NOLOGGING]
    [DEFAULT storage_clause ]
    [ONLINE|OFFLINE]
    [PERMANENT|TEMPORARY]
    [extent_management_clause]
    [segment_management_clause]
```

Creating Tablespaces

Creating Tablespace using SQL