Data Base Administration

Dr. B. B. Meshram

1 Oracle Database Files

An oracle database is a collection of data that is treated as a unit. The database has a logical structure and a physical structure. The physical structure of the database is the set of operating system files in the database. An oracle database consists of three file types: Data files, Redo logs, control files. The oracle server uses other key physical structure that are not the part of the database. There are the parameter file, password files, and Archived redo log files.[3,4,34,40]

Parameter file: The parameter file is read by either SQL*Plus, Server Manager, or Instance Manager when you use one of those programs to start an instance. Without the correct parameter file, you may not be able to open your database. Parameter file contain a number of settings that influence how an oracle database instance functions. Some of more important aspects that you can control via settings in the parameter file are the following

The location of the database control files.

- The amount of memory oracle uses to buffer data that has been read from disk.
- The amount of memory oracle uses to buffer SQL statement execution plan, PL/SQL procedures, and data dictionary information.
- The default optimizer choice.

Control files: Oracle uses control files to store information about the state of your database. Database control files contain the following major types of information about

- The database name,
- Information about tablespace, the names and locations of all the redo log files, the current log sequence number, Chelkpoint information, Information about redoes logs and current state of archiving.

Control files are so important and so critical, that oracle recommends that you always mirror your control files –maintaining two or three copies so that if one is lost, you have the others to fall back on.

Data files: Oracle stores your data in data files. Data file contents are table data, Index data, data dictionary definitions, and Information necessary to undo transactions (rollback data), code for stored procedures, functions and packages, and temporary data.

Log files: Log files known, as redo log files are an important components of any database. Log file contain a sequential record of changes to a database. Any time you execute a SQL statement that changes the data in your database, oracle generates one or more redo log entries to record that change. They exist to ensure the recoverability of a database in the event of a system crash, a drive failure, or any other unforeseen interruption to normal operations. There is two types of redo log files: On line redo log files and offline redo log files (Archived redo log files). All oracle database use on line redo log files., Archived redo log files are only generated when a database is running archive log mode, and allow for up-to-the-minute recovery in the event that a data file is lost

. The password file authenticates which users are permitted to start up and shut down an oracle instance.

2 Managing An Oracle Instance

An oracle instance is not available to users until the database administrator has started the instance and opened the database. Every time an instance is started, oracle uses a parameter file, which contains initialization parameters, to allocate SGA and to start the back ground processes. The parameter file, commonly referred to, as the initsid.ora file is a text file that can be maintained using standard operating system editor. The parameter file is as in **appendix A.**. The server has a default value for each parameter. This value may be operating system dependent, depending on the parameter and some parameters need to be modified as per the requirements.[1,2,3,4]

To set an operational environment, apply the following steps:

1. Create folders: datafiles, control files, logfiles in c:\oracle
2. Make the required changes in the init.ora file and after making changes save it as “initbandu.ora” it may be any significant file name.

The following changes can be made in init.ora file

```
db_name=bandu
Instance_name=bandu
Servilce_name=bandu
Control files=(“c:\oracle\datafile\controlfile\ control01.ct1”,”
               c:\oracle\datafile\control control02.ct1”)
Ult_file_div=c:\datafile
```

Creating A Database Manually

The following steps are used to create a database manually.

Password File

1, On windows NT

- Set the variable Oracle SID to use SQL* plus with the command

```
C:/>Set ORACLE-SID = bandu
```

- Create the service and the password file with ORADIM utility with the command

```
C:> ORADIM – NEW – SID = bandu  - INTPWD bandu  - STARTMODE AUTO – PFILE = C:\DATAFILE\initbandu.ora
```

To delete password file the service created, use the command:

```
C:\>ORADIM-DELETE-SID<Sidname>
```

For creating a password file, properly set the REMOTE-LOGIN-PASSWORD FILE initialization parameter you can set three possible values namely none, shared and exclusive for the parameter.

2.4.1.2 Editing The Parameter File

When preparing the new database, copy the default init.ora file with the new name init SID.ora say initBandu.ora and change the settings for some parameter as shown; others can be left to default. You should specify at least the following parameters before starting the instance: DB-NAME, CONTROL-FILES DB-BLOCK-SIZE

CREATING THE DATABASE

To create a database use the following script

```
CREATE DATABASE BANDU
MAXLOGFILES                         32
MAXLOGMEMBERS                       05
MAXDATAFILES                        100
MAXLOGHISTOPY                       100
LOGFILE
GROUP1('C:\ oracle\datafile\ Controlfile\log1a.rdo'
                  'C:\oracle\datafile\controlfile\log1b.rdo') SIZE 1M
GROUP2   ('C:\oracle\datafile\dadtafile\controlfile\log2a.rdo',
                  'C:\oracle\datafile\controlfile\log2b.rdo') SIZE 1M
DATAFILE
'C:\oracle \datafile\ sysbandu.dbf ' size 50m autoextend on
'C:\oracle\ datafile\  app-data01.dbf  '        size 50m autoextend on
'C:\oracle\datafile\opp.data.o2 .dbf ' size 50m autoextend on
CHARACTER SET US7ascii;
```

See more details of the CREATE DATABASE command in [1,2,3] After the database is created the database is opened, the SQL script SQL.bsq is successfully executed.

You can view the dynamic performance views such as V$LOGFILE, V$ CONTROL FILE and DATAFILE, but no data dictionary views are created.

Starting The Instance

Connect as SYSDBA using operating system authentication or the password file method and start the instance using the STARTUP commond as given below :[1,2,3,4,34,36]

```
SQL>connect internal/bandu
SQL>. Startup no mount pfile=c:\datafile\initbandu.ora
```

Due to this commond you will get the following output

```
Oracle instance started
Total system global Area                  36437964 byte
Fixed size                              65484 byte
Variable size                           19521536 bytes
Database buffer                         16777216 bytes
Redo buffers                            73728 bytes

Database mounted
Database opened
```

3.Creating The Data Dictionary

The base tables of the data dictionary are automatically created when a database is created. When the database is created with the oracle universal Installer, the data dictionary views and scripts for your oracle server options are run automatically, you might need to run them again when upgrading to new release of the oracle server.[1,3,18,22,24,26,30] as below:

```
C:\>Svrgmrl   [ press enter]
Svrmgrl>@C:\oracle\ora8i\rdbms\admin\cat*.sql
Svrngrl>@C:\oracle\ora8i\rdbm\admin\dbms*.sql
Svrngrl>@ C:\oracle\ora8i\rdbm\admin \ prvt*.plb
Svrngrl>@ C:\oracle\ora8i\rdbm\admin \utl*.sql
```

5 Managing Tablespace And Data Files

An oracle database can be logically grouped into smaller logical area of space known as tablespace. Each tablespace consists of one or more operating system files, which are called datafile. A segment is the space allocated for a specific logical storage structure within a tablespace for example, all of the storage allocated to a table is a segment. Space is allocated to a segment by extents. One or more extends make up a segment.[1,2,3,4,,31,33]

Creating tablespace

You create a teblespace with the CREATE TABLESSPACE command:

```
CREATE TABALESPACE table space
DATAFILE detafile_clause]   [. Detafiile_clause]… [MINIMUM EXTENT integer [K[M]]
```

[LOGING/OFFLINE] [DEFAULT storage_clause] [ONLINE/OFFLINE] [PERMANENT/TEMPORAY] [Extent_management_clause]

More details of the command, for creating tablespace are discussed in [1,3,30,31 18,22,24,26]

For example:

```
CREATE TABLESAPCE app-data
DATAFILE  'C:\oracle\app_datao1.dbf' size   100 M
              "C:\oracle\app_data02.dbf' size 100M
 MINMUM EXTENT 500K
 DEFAULT STORAGE (INITIAL 50K, NEXT 500K, MAXEXTENTS 500, PCTINCREASE 0);
```

Changing Default Storage Settings

Use the ALTER TABLESPACE command to alter the default storage definition of a tablespace. For example:

```
ALTER TABALESPACE tablespace
[MINIMUM EXTENT integer [K|M]
[DEFAULT storage_clause}
```

For Example: -

```
ALTER TABLESPACE app_data
DEFAULT STORAGE (INITIAL 3M, NEXT 3M, MAXEXTENTS 900);
```

Taking tablespace offline or online

To take a tabalespace offline use the command:

```
ALTER TABLESAPCE app_data OFFLINE;
```

To bring a tablespace online:

```
AFTER TABLESPACE app_data ONLINE;
```

Read-only tablespace

Taking tablespace read-only prevents further write operations on the datafile in the tablespace. Read-only tables spaces eliminate the need to perform backups of large, static portions of a database.

```
ALTER TABLESPACE tablespace[ READ ONLY | WRITE]
```

The detailed discussion of this command is presented in[2,3,4,33]

For Example :

```
ALTER TABLESPACE app_data READ-ONLY
```

To make a read-only tablespace writ able, all of the datafile in the tablespace must be online.

Drop tablespace

```
DROP TABLESPACE tablespace
[[INCLUDNG CONTENTS [CASCADE CONSTRAINTS]]
```

For detailed discussion of definition of this command the reader is directed to [33,34]

Enabling Automatic Resizing Of DATAFILES

You can enlarge a tablespace in to ways :

1.Change the size of a datafile   2. Add a datafile to a tablespace

Add a datafile to a tablespace

```
ALTER TABLESLPACE tablespace ADD DATAFILE filespace [autextened _clause]
Filspace  [autoextended_change]
```

The detailed discussion of this command is presented in[2,3,4,33]

For example:

```
ALTER TABLESAPCE app_data
ADD DATAFILE ‘C:\oracle\app-data-04.dbf’
SIZE 200M
AUTOEXTND ON
NEST 10M
MAXSIZE 50M ;
```

Change the size of a datafile

Instead of adding space to the database by adding datafile, the DBA can change the size of the datafile

For Example:

```
 ALTER DATABASE
DATAILE ‘C:\appdata.dbf’
RESIZE 200M;
```

You can add datafile to a tablespace to increase the total amount of disk space allocated for the tablespace with the ALTER TABLESPACE ADD DATAFILE commond

```
ALTER TABLESPACE tablespace
ADD DATAFILE filespace  [autoextend.clause]
[filespace[autoextend_caluse]]……..
```

The detailed discussion of this command is presented in[2,3,4]

For example:

```
 ALTER TABLESPACE app-date
ADD DATAFILE ‘C:\oracle\app-data-03.dbf’
Size 200M;
```

5 Managing Rollback Segments

The rollback is part of transaction Recovery. Recovery is only possible because changes made to the rollback segment are also protected by the redo log files.

There are three types of rollback segments

1.SYSTEM rollback segment, 2.NON-SYSTEM rollback segment; 3.Deferred rollback segments

The detailed discussion of this is presented in[1,2,3,22,24]

1.Use the following command to create a rollback segment

```
CREATE [PUBLIC] ROLLBACK SEGEMENT rollback_segment [TABLESPACE
tablespace][STORAGE ([INITIAL integer [K|M] [MINEXTENTS integer]
MAXETENTS integer|UNLIMITED] [OPTIMAL integer [K| M] |NULL]
```

For detailed discussion of definition of this command the reader is directed to [2,4]

For Example –

```
CRATE ROLLBACK SEGMENT rbs01
TABLESPACE rbs
STORAGE (
INITIAL 200k
NEXT 200k
MINEXTENTS 40
MAXEXTENTS 200
OPTIMAL 4000k);
```

2. When a rollback segment is created, it is offline and can not be used. The following command is used to make a rollback segment available:

```
ALTER ROLLBACK SEGMENT rollback_ Segment
[STORAGE (NEXT integer [K|M]] [MINEXTENTS integer] [MAXEXTENTS
integer|UNLIMITED] [OPTIMAL {integer [K/M]|NULL}]
```

For detailed discussion of definition of this command the reader is directed to [3,4,31]

3. To ensure that a rollback segment is always brought online by an instance, specify the name of the rollback segment in the parameter file as shown.

```
ROLL BACK_SEGEMENTS=[rbo1, rbo2]
```

4.The storage parameters for a rollback segment can be changed using the ALTER ROLLBACK SEGMENT command.

```
ALTER ROLLBACK SEGMENT rollback_segment
[STORAGE [[NEXT integer [K/M]][Min]
[MINEXTENTS integer]
[MAXEXTENTS {integer/UNLIMITED}]
[OPTIMAL {integer [K/M}]
```

For detailed discussion of definition of this command the reader is directed to [3,4,18]

For Example :

```
ALTER ROLLBACK SEGMENT rbs01
STORAGE (MAXEXTENTS 300);
ONE LINE ;
```

5. The following command attempt to reduce the size of the rollback segment to the specified size:

```
ALTER ROLLBACK SEGMENT SHRINK [To integer [K | M];
```

6. To take a rollback segment offline use the command:

```
ALTER ROLLBACK SEGMENT rollback-segment offline;
```

For Example

```
ALTER ROLLBACK SEGMENT rbso1
```

7. A rollback segment must be offline before it can be dropped. Also take a rollback segment offline to prevent new transactions from using a rollback segment.

```
DROP ROLLBACK SEGEMENT rollback_ segment;
```

For example

```
DROP ROLLBACK SEGMENT rbso1;
```

8.To obtain information about all the rollback segments, issue the following commond:

```
SQL>SELECT segment_name, tablespace_name, owner, statues
FROM dba_rollback_segs;
```

MANAGING TABLES, MAINTAINING DATA INTEGRITY and LOADING DATA are the common general commonds known to oracle developers and For detailed discussion of definition of these concepts the reader is directed to [1,2,3,4,5,,18,22,24,30,31]

6 Managing Users

This section explores how you can use SQL statements to create users and roles, assign privileges and control passwords.

Profile

Creating profile [passwords setting]

A profile is a collection of parameters given a name and assigned to one or more oracle users.[26,,34,40] Use the following CREATE PROFILE command to administer passwords :

```
CREATE PROFILE profile LIMIT [FAILED_LOGIN_/ATTEMPTS max_value],
[PASSWORD_LIFE_TIME max_value], [{PASSWORD_RESUE_TIME |
PASSWORE_RESUME_MAX max_value] [ACCOUNT_LOCK_TIME max_value],
[PASSWORD_GRACE_TIME max_value} [PASSWORD_VERIFY_FUNCTION
[function/NUL/DEFAULT]
```

For Example :

```
CREATE PROFILE sonameshram LIMIT
FAILED_LOGIN_attempts 3
PASSWORD_LOCK_TIME UNLIMITED
PASSWORD RESUE_TIME 30
PASSWORD_LILFE_TIME 30
PASSWORD_VERIFT_FUNCTION verify_function
PASSWORD_GRACE_TIME 5;
```

Altering and dropping a profile.

ALTER PROFILE profile LIMIT [FAILED_LOGIN_ATTEMPTS max_value][PASSWORD_LIFE_TIME max_value][{PASSWORD_RESUE_IME| PASSWORD_RESUE_MAX max_value][ACCOUNT_LOCL_TIME max_value][PASSWORD_GRACE_TIME max_value][PASSWORD_VERTIFY_FUNCATION max_value]

Dropping a profile

DROP PROFILE profile [CASCADE]

For Example : DROP PROFILE developer_prof CASCADE ;

CREATING a Profile : Resource limit

Create of profile using the following CREATE PROFILE command:

CREATE PROFILE profile limit
[SESSION_PER_USER max_value][CPU_PER_SESSION max_value][CPU_PER CALL max_value][CONNECT_TIME max_value][IDLE_TIME max_value][LOGICAL_READS_PER_SESSION max_value][LOGICAL_READS_PER_CALL max_value][COMPOSITE_LIMIT max_value][PRIVATE_SGA max_bytes]

For Example :

```
CRATE PROFILE developer_proof LIMIT
SESSION_PER_USER 2
CPU_PER_SESSION 10000
IDLE_TIOME 60
CONNECT_TIME 480;
```

For detailed discussion of profile commands, the reader is directed to [1,3,34]

Creating users

For Example:

```
CREATE USER trsontakke
IDENTIFIED By sontakkemeshram
DEFAULT TABALESPACE   app_ data
TEMPORARY TABLESPACE temp
QUOTA 15M ON    app_  data
PASSWORD EXPIRES;
```

Creating a new user : operating system Authentication :

Use the INDENTIFIED EXTERNALLY clause of the CREATE USER command to specify that a user must be authenticated by the operating system.

In init.ora, the OS-AUTHENT-PREFIX initialization parameter is used to specify the format of the username for the operating system authentication. This value default to ops & to set the preix to a null value.

OS_AUTHENT_PREFIX =   “   “

Using OS-AUTHENT-PREFIX=OPS$ gives the flexibility of having a user authenticated by the operating system of the oracle server. In this case, the DBA can create the user by entering the command of the form:

CREATE USER OPS$ user
IDENTIFIED BY password ……

DBA can set another initialization parameter, REMOTE_OS_AUTHENT=True to authenticate the user by a remote operating system.

Modify tablespace .DBA use the following command to modify tablespace quotas or to reassign :

ALTER USER USER [DEFAULT TABLESPACE tablespace][TEMPORAY TEBALESPACE tablespace][QUOTA {inter [K | M |UNLIMITED} ON tablespace [QUOTA {integer [K | M | UNLIMITED] ON tebalespace].]

Dropping user :

DROP USER Username;
DROP USER Username CASECDE;

The CASCADE option drops all objects in the schema before dropping the user

The DBA_USERS view provides information about the users that exist in your database by using the following query.

SQL>SELECT username, account, status from dba-users
ORDER by username;

For detailed discussion of 'Managing Users' commonds, the reader is directed to [3,11]

Granting Roles And Privileges To Users

For a user to do anything, DBA need to grant the user one or more system privileges. A system privilege is a privilege defined by the oracle that allows a user to perform a certain task, such as creating a table. Privileges are frequently bundled together into roles.

There are approximately 126 system privileges. [3,11,15,18]

system privilege : To grant a system privilege or a role to a user, use the GRANT Command as follows:

GRANT {system_priiv|role} [,{system_priv|role}] TO {user | role | PUBLIC} [, {user | role |PUBLIC}] [WITH ADMIN OPTION]

For Example :

GRANT CREATE SESSION TO SCOTT WITH ADMIN OPTION;

Commonly granted privilege are create session, connect, resource, create table, create view, create sequence, create procedure, create trigger, create synonym.

You can list as many as privileges and roles in the command as you need.

For Example.

GRANT CREATE SESSION, CREATE TABLE, CONNECT, UNLIMITED TABLESPACE TO Joseph;.

By default, only DBA can grant roles or privileges to other users. If you happen to have a specific role or privileges that you want someone else to manage for you, you can grant it to him or her using the ADMIN option.

For Example:

GRANT CREATE TABLE to Joseph WITH ADMIN OPTION;

For Example:

To give all privileges to a user say bbmeshram

SQL> create user bbmeshram IDENTIFIED by sonameshram;

SQL> grant connect, resource to bbmeshram;

Due to this command all privileges are granted to bbmeshram by DBA.

Revoking system privileges and roles:

You can use the REVOKE command to remove a role or privilege from the user. The syntax is :

REVOKE {system_prive | role} [, {system_prive | role}]. FROM {user|role|PUBLIC} [, {user | role | PUBLIC}]…………

For Example:

The following command revoke create table from the user named josheph.:

REVOKE CREATE TABLE FROM Josheph,

If you have granted a privilege to PUBLEC you can revoke it by using PUBLIC in place of the username.

For Example.

REVOKLE UNLIMITED TABLESPACE FROM PUBLIC. ;

As with a GRANT statement, you may revoke any number of roles and privileges using one command.

Granting object privileges

Use the following command for Granting object privileges

GRANT {object_priv [{column_list}][, object_prive[(column_list)]. | ALL [PRIVILEGES}] ON [schema.]object TO {user |role|PUBLIC} [, {user|role|PUBLIC}]. [WITH GRANT OPTION]

Revoking object privileges

REVOKE {object_priv [, object_priv].|[PRIVILEGES]} ON [schema.]object FROM {user|role|PUBLIC} [, {user|role|PUBLIC}].[CASCADE CONSTRAINTS]

creating a role

You use the following command to create a role

CREATE ROLE role[NOT IDENTIFIED | IDENTIFIED {by password | EXTERNALLY}]

Modifying roles

ALTER ROLE role {NOT IDENTIFIED | IDENTIFIED {by password | EXTERNALLY }];

Enabling and disabling roles

SET ROLE {role [IDENTIFIED BY PASSWORD] [, role [IDENTIFIED BY PASSWORD]]…|ALL [EXCEPT role [, role].]| NONE}

For detailed discussion of 'granting roles and previledges to users' commonds, the reader is directed to [3,4]

8 Using Fine Grained Access Control

An exciting new oracle 8i feature is the ability to implement row-level security within the database. Oracle referees to this as fine-grained access control.

An application context is a scratch pad area in memory that you can use to store bit of information that security policies need. For example, When a user logs on, you might store his or her user name, ID number, department number, and other information in an application context.

Creating a context. The syntax for the create context command looks like this.

CREATE [OR REPLACE] CONTEXT Context-name USING [schema.] package ;

The package name that you supply when you create a context identifies the PL/SQL package that is allowed to create and set attributes within the context. The package does not need to exist when you create the context. You can create it afterwards.

Dropping a context: You use the Drop Context command to remove a context.

DROP CONTEXT context-name;

Define attribute in context: Once you have created a context, you can define attribute in that context using the DBMS_SESSION PACKAGE.The syntax of that call looks like this: DBMS_SESSION.SET_CONTEXT('context-name','attribute-name',' 'attrtibute.value');

For example, the records of animals and tanks may be modified only by the care taker responsible for the animal or tank in question. if you had a escapable application user responsible for tank*4, and you wanted to store that information in the SEAPARK context, you could call SET-CONTEXT like this:

DBMS-SESSION.SET.CONTEXT("SEAPARK", 'TANK', '4');

Retrive attrbute from a context

You can retrive attribute values from a context using oracle's new built-in SYS-CONTEXT function. The syntax for calling that function is as follows.

Attribute-value:=SYS-CONTEXT('Context-name', 'attribute-name');

For example:

```
SQL>Select SYS.Context ('seamark', 'username') from dual;
SYS-CONTEXT ('SEAPAK', 'USERNAME');
 bbmeshram
1 row selected.
```

9 Writing Security Policies

To implement a policy, you generally follow these steps :

1. Write a stored function or a package that conforms to the policy specification.

2 Use the DBMS-RLS.ADD-POLICY procedure to associate the policy function with the tables that you are protecting

.Policy is written as PL/SQL functions. Oracle call these functions automatically and expects them to conform to this specification.

FUNCTION function-name(Object-schema IN VARCHAR2 Object-name in VARACHAR2) RETURN VARACHER2;

You use the DBMS-RLS.ADD.POLICYY procedure to link a policy function with a table. The syntax looks line this :

DBMS.RLS.ADD.POLICY
(.object_schema' object_name, Policy_name function_schema, Function name, statement types, update-check)

Dropping Policy

You can drop (remove) a policy by making a call to the DBMS-RLS packages DROP-POLICY procedure. The syntax for this looks like the following :

DBBMS-RLS.DROP-POLICY
(Object-Schema,
Object –name,
Policy-name);

People have used a few approaches in the past to implement row, level security within the database, These approaches generally include some combination of the following :

- Views to restrict the data that a user can see
- Triggers to prevent unauthorized updates to tables
- Stored procedures as a vehicle for managing updates and deletes

For detailed discussion of 'fine grained access control' , the reader is directed to [1,2,3,4,5,11,15,18,22,30,31,]

APPENDIX A: Sample initbandu.ora file

```
db_name = BANDU
instance_name = BANDU
service_names = BANDU
db_files = 1024
control_files = ("C:\ORACLEHOME\DATAFILE\CONTROLFILE\control01.ctl",
"C:\ORACLEHOME\DATAFILE\CONTROLFILE\control02.ctl", )
db_file_multiblock_read_count = 8
db_block_buffers = 1028
shared_pool_size = 52428800
```

```
java_pool_size = 20971520
log_checkpoint_interval = 10000
log_checkpoint_timeout = 1800
processes = 50
parallel_max_servers = 5
log_buffer = 32768
#audit_trail = true  # if you want auditing
#timed_statistics = true  # if you want timed statistics
max_dump_file_size = 10240  # limit trace file size to 5M each
# For archiving if archiving is enabled #
log_archive_start = true
 log_archive_dest_1 = "location=C:\Oracle\oradata\wen\archive"
log_archive_format = arc%S.%T
# Global Naming -- enforce that a dblink has same name as the db it connects
to
global_names = true
oracle_trace_enable = true
oracle_trace_collection_name = ""
# define directories to store trace and alert files
background_dump_dest = C:\Oracle\admin\wen\bdump
user_dump_dest = C:\Oracle\admin\wen\udump
db_block_size = 2048
remote_login_passwordfile = exclusive
os_authent_prefix = ""
# The following parameters are needed for the Advanced Replication Option
job_queue_processes = 0
job_queue_interval = 60
open_links = 4
distributed_transactions = 10
mts_dispatchers = "(PROTOCOL=TCP)(PRE=oracle.aurora.server.SGiopServer)"
# Uncomment the following line when your listener is configured for SSL
# (listener.ora and sqlnet.ora)
# mts_dispatchers =
"(PROTOCOL=TCPS)(PRE=oracle.aurora.server.SGiopServer)"
mts_servers = 1
compatible = 8.1.0
sort_area_size = 66560
```

APPENDIX B: PARAMETERS OF PARAMETER FILE

An initialization parameter file is an ASCII text file containing a lest of parameters. Some parameter are listed below.

Initialization parameter name	Definition	Default value	Range of values
INSTANCE_NAME	Representing the name of the instance	SID	Any alphanumeric characters.
SERVICE_NAMES	Specifies the service names supported by the instance.	DB-NAME.	Any ASCII strong.
DB_BLOCK_BUFFERS	Specifies the number of database buffers available in the buffer cache.	50	Operating system-specific.
DB_BLOCK-CHECLPOINT_BATCH	Specifies the number of buffers that will be added to each batch of buffers that DBWR writes in order to advance checkpoint processing.	8	0-derived.
DB_BLOCK_CHECKING	It is used to control whether block checking is done from transaction management blocks.	FALSE	FALSE, TRUE.
DB_BLOCK-CHECKSUM	DBWn and the direct loader will calculate a checksum and store it in the cache header of every data block when writing it to disk.	FALSE	FALSE, TRUE.
DB-BLOCK_SIZE	Specifies the size in bytes of Oracle database blocks.	Operating system dependent	2048-32768.
DB-WRITER_PROCESSES	Specifies the initial number of database writer processes for an instance.	1	10

DBWR_IO-SLAVES	Specifies the number of I/O slaves used by the DBWR process.	0	0 to system depending value.
DISK_ASYNCH_IO	Used to control whether I/O to datafilse, controlfiles and logfiles are asynchronous.	TRUE	TRUE, FLASE
DML_LOCKS	Specifies the maximum number of DML locks-one for each table modified in a transaction.	Derived (4*TRANSACTIONS)	20- unlimited.
SHARED_POOL_SIZE	Specifies the size of the shared pool in bytes.	3,500,000 bytes.	300 Kbytes.
LOG-ARCHIVE_BUFFER-SIZE	Specifies the size of each archival buffer, in redo log blocks.	Operating system-dependent.	1-operating system blocks.
LOG-ARCHIVE-BUFFERS.	Specifies the number of buffers to allocate for archiving.	Operating system-dependent.	Operating system-dependent.
LOG-ARCHIVE-DEST	Applicable only if you are using the redo log in ARCHIVELOG mode.	Operating system-dependent.	Operating system-dependent.
MAX-DUMP_FILE_SIZE	Specifies the maximum size of trace files to be written.	100000 blocks.	0-unlimited.
SQL_TRACE	Disables or enables the SQL trace facility.	FALES	TRUE/FALSE.
USER_DUMP-DEST	Specifies the pathname for a directory where the server will write debugging trace files on behalf of a user process.	Operating system-dependent.	Valid local pathname, directory, or disk.
OPTIMIZER_MODE	Specifies the	CHOOSE	RULE/CHOOSE/

	behavior of the optimizer.		FIRST_ROWS/ ALL_ROWS
ARCH_IO_SLAVES	Specifies the number of I/O slaves used by the ARCH process to archive redo logfiles.	0	0-15.
BACKGROUND_DUMP_DEST	Specifies the pathname for a directory where debugging trace files for the background process.	Operating system-dependent.	Valid local pathname, directory or disk.
BACKUP_DISK_IO_ SLAVES	Specifies the number of I/O slaves used by the Recovery manager to backup, copy, or restore.	0	0-15
BACKUP-TAPE_IO-SLAVES	Specifies whether I/O slaves are used by the Recovery manager to backup, copy or restore data to tape.	FALSE	TURE/FALSE

.

REFERENCES

1. Thomas, Biju, Stuns, Doug, OCP: Oracle8i DBA Architecture & Administration & Backup & Recovery Study Guide, Ocp: Oracle8i Study Guides, Sybex incorporated, 2000.
2. CIP Staff Orcle8i DBA Architecture & Administration, Exam Cram Ser, Coriolis Group, LLC, June - 2001.
3. Bruce Ernst, Hanne Rue Rasmussen et.al, Enterprise DBA Part 1A, Architecture And Administration, Volum1, Student Guide, Sherry Polm, Oracle, August 1999.
4. Bruce Ernst, Hanne Rue Rasmussen et.al, Enterprise DBA Part 1A, Architecture And Administration, Volum1, Student Guide, Sherry Polm, Oracle, August 1999.
5. Johnson, Joe, Dawes, Chip, Weishan, Matthew, OCP: Oracle8i DBA Certification Kit, Sybex. Incorporated, December 2000.

6. Johnson, Joseph C., Weishan, Matthew, OCP: Oracle8i DBA Performance Tunnin and Network Administration Study Guide, Ocp: Oricale8i Study Guides, Sybex, Incorporated, August 2000.
7. Dominique Jeunot, Shankar Raman, Enterprise DBA part 1B: Backup And Recovery Workshop, Volume 1, Publisher Johan Dawson, Oracle, March-1998.
8. Dominique Jeunot, Shankar Raman, Enterprise DBA part 1B: Backup And Recovery Workshop, Volume 2, Publisher Johan Dawson, Oracle, March-1998.
9. Dawes, Chip, Thomas, Biju OCP: Oracle8i DBA SQL and PL/SQL Study Guide, Sybex, Incorporated, September-2000.
10. CP: Oracle8i DBA Virtual Test Center, Sybex ~ Trade Paperback Mar. 2001.
11. Loney Kevin, Theriault, Marlene, Oracle8i DBA Handbook, McGrawHill Professional Book Group, October 1999.

12. Gennick, Jonathan Oracle8i DBA's Bible Ser ~ Trade Paperback Hunger Minds, March 2000.
13. Couchman, Jason S. Oracle8i Certified Professional DBA Practice Exams, Editimz, MCGrawHil, Professional Book Group, April 2001.
14. Counchman, Jason S., Oracle8i Certified Professional DBP upgrade Exam Guide, MCGrawHil Professional book Group, June 2001.
15. Counchman, Jason S., Schwinn, Ulrike, Oracle8i Certified professional DBA Certification Exam Guide (With CD-ROM), MCGrawHil Professional Book Group, June 2001.
16. Dominique Jennot, Enterprise DBA Part2: Performance And Tuning Volume 1 student Guide, Publisher Kelly Lee, Oracle Press, September 1999.
17. Dominique Jennot, Enterprise DBA Part2: Performance And Tuning Volume 2 student Guide, Publisher Kelly Lee, Oracle Press, September 1999.
18. Lane, Pual, Topper, Bill, Oracle 8 – 8i DBA Certification, Oracle Bks, February – 2000.
19. Jason S. Couchman, Oracle8i Certified Professional SQL & PLSQL Exam Guide, Oracle press books, 1999.
20. Jason S. Couchman, Oracle8i Certified Professional DBA Upgrade Exam Guide, 1999.
21. Jonathan Gennick, Carol McCullough-Dieter et.al, Oracle8i DBA Bible, Idg Books India (P) Ltd. 2000.
22. George Koch, Kevin Loney, Oracale8 The complete Reference, Oracle Press, Tate McGraw-Hill, 1998.

23.Sumit Sarin, Oracle DBA Tips and Techniques, Osborne/Mc-Graw-Hill, 2000.
24.Kevin Loney, George K. Koch, Oracle81, The complete Reference, Osborne McGraw-Hill (Oracle Press) May 23, 2000
25.Venkat S. Devraj, Oracle 24x7 Tips & Techniques Computing McGrraw-Hill (Oracle Press)November 1999.
26.Kevin Loney, George K. Koch, Oracle8-The Complete Reference, Oracle Press, September 1997
27.Jonathan Lewis, Practical Oracle 8i Building Efficient Databases, Oracle Prass 2000
28.Rama Velpuri, Anand Adkoli, Oracle8 Backpu & Recovery Handbook (Oracle Press|)
29.Grag Spence, Sap R/3 and Oracle Backup and Recovery
30.Rick Greewald, Jonathan Stem, Robert Stackowiak, Oracle Essentials, Oracle8 and Oacle8i, O'Reilly & Associates, Novembe, 1999
31.Kevin Loney, Mariene L. Theriault, Oracle8i DBA Handbook, Osbrone McGraw-Hill (Oracle Press)October 12, 1999
32.Steven Feuerstein, Charies Dye, Johan Beresniewica, Oracle Built-in Packages, O'Reilly & Associates, May 1998
33.Ulrike Schwinn, Jason S. Couchman, Jeremy Judson, Oracle 8i,Certified Professional DBA Certification Exam Guide, Osborne McGraw-Hill (Oracle Press), December15, 2000
34.Doug Stuns, Biju Thomas, Greg Hobbs, OCP Oracle8i DBA Architecture Administration and Backup & Recovery Study guide, Sybex, November, 2000
35.Joseph C. Johnson, Mathew Weishan, Joe Johnson, OCP Oracle8i DBA Performance tuning and Netwok Administration Study Guide, Sybex, September, 2000
36.Chip Dawes, BijuThomas, OCP Oracle8i DBA Sql and PL/SQL Study Guide, Exam 1Z0-001, Sybex, September, 2000
37.Mark Gurry, Peter Corrigan, Oracle Peformance Tuning, O'Reilly & Associates, November, 1996
38.Eyal Aronoff, Kevin Loney, Noorali Sonawalla, Oracle8 Advanced Tuning & Administration, Osborne McGraw-Hill (Oracle Press) June9, 1998
39.Michael J. Corey (Editor), Michael Abbey, Daniel J. Jr. Dechichio, Abbott Corey, Ian Abramson, Oracle 8i tuning,December 1997
40.Steve Bobrowski, Oracle8 Architecture, Osoborne McGraw-Hill (Oracle Press) October 1997
41.Eyal Aronoff, Kevin Loney, Noorali Sonawalla, Oracle8 Advanced Tunig & Administration, Osbrone McGraw-Hill (Oracle Press) June 9, 1998

42.Andy Tremayne, Oracle Applications Performance Tuning Handbook (Book/CD-Rom Package),Osborne McGraw-Hill (Oracle Press), September 19,2000.