

Getting Familiar with Oracle Database 12c

Duration: 45 Minutes

Prerequisites For the Session – Oracle Database Enterprise edition should be installed and configured with ORCL database.

Activity 2.

Connecting to Oracle Database and Building an Awareness on Oracle Instance

Open a terminal on Linux and on the shell continue as below.

```
[root@localhost Desktop]# su - oracle
```

```
Last login: Thu Sep 7 01:30:04 EDT 2017 on pts/0
```

```
[oracle@localhost ~]$
```

Use the Echo command to check the current value of ORACLE_SID variable

```
[oracle@localhost ~]$ echo $ORACLE_SID
```

```
orcl
```

```
[oracle@localhost ~]$
```

As explained in Activity 1 Dollar - \$ in front of the environment variable ORACLE_SID recalls the value of the variable and echo command will display it.

As per the output current value is 'orcl', this indicates the value of the ORACLE_SID for the current Oracle user terminal is orcl.

This means the terminal is ready to connect to database called 'orcl'.

SID – System identification, which is setting a unique value which represents the database name.

How to set the value for the variable?

```
[oracle@localhost ~]$ export ORACLE_SID=abc
```

The variable name has to be in capital in Linux, UNIX and in Solaris.

The value is the Db name which is in simple case here.

The variable can hold only one value at a given time, in case of above, now it is ready to connect to a database called 'abc'.

If no value is set, the value of the .bash_profile is taken.

export PATH

Oracle Settings

export TMP=/tmp

export TMPDIR=\$TMP

export ORACLE_HOSTNAME=localhost.localdomain

export ORACLE_UNQNAME=orcl

export ORACLE_BASE=/u01/app/oracle

export ORACLE_HOME=\$ORACLE_BASE/product/12.1.0/dbhome_1

export ORACLE_SID=orcl

Let's connect to the database.

[oracle@localhost ~]\$ sqlplus sys as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Tue Sep 12 05:15:06 2017

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

Enter password:

Connected to an idle instance.

sys – is the super user for the oracle database like root for Linux OS.

Here no password given, then the OS authentication will take place. The OS user must be an administrative user. This is the default behavior of the database which cause a security issue as the SA – system admin have the super user access for the database.

It is recommended to disable this in production environment and introduce database level authentication which will then require a password for the authentication. This will be described below in this guide.

Sysdba – this is a special role which is built in to the database and if the user is sys it is a must to use the role.

Finally “**Connected to an idle instance.**” This means the database is down or not running.

The reason is that SID value is ‘abc’, and there is no such database running now.

If a database is up and running the message “**Connected to an idle instance.**” Won’t be there.

The output will be as follows.

```
[oracle@localhost ~]$ sqlplus sys as sysdba
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Tue Sep 12 05:26:48 2017
```

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

```
Enter password:
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing  
options
```

If the output is “**Connected to an idle instance.**” Then to start the DB, issue a startup command.

```
[oracle@localhost ~]$ export ORACLE_SID=abc
```

```
[oracle@localhost ~]$ sqlplus sys as sysdba
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Tue Sep 12 05:30:34 2017
```

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

```
Enter password:
```

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

```
SQL> startup
```

```
ORA-01078: failure in processing system parameters
```

```
LRM-00109: could not open parameter file
```

```
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/initabc.ora'
```

The error message indicates that there is no parameter file exists , this confirms that the value of the ORACLE_SID is not correct.

Connecting to oracle database continues...

[oracle@localhost ~]\$ export ORACLE_SID=orcl

[oracle@localhost ~]\$ sqlplus / as sysdba

SQL*Plus: Release 12.1.0.2.0 Production on Tue Sep 12 05:34:12 2017

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

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

Above is another method which directly goes through OS authentication even without prompting the password. The user connected in both cases will be 'sys'.

SQL> show user

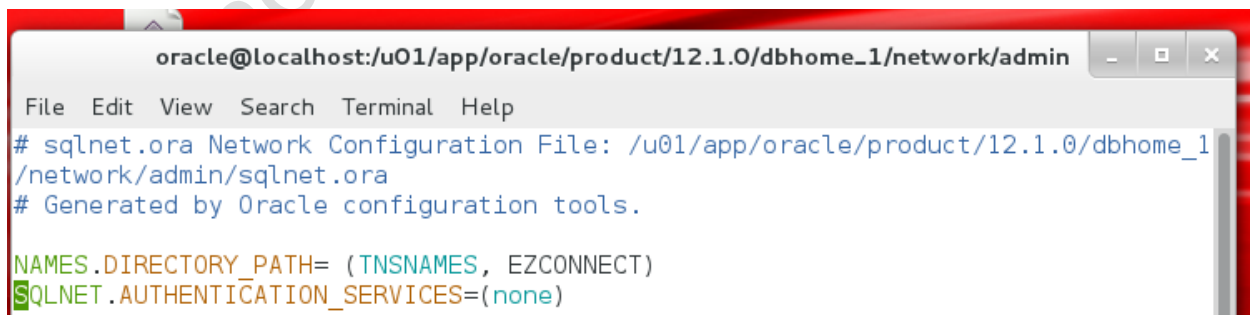
USER is "SYS"

SQL>

Please note

Connecting as super user that is 'sys' is extremely for database administrative purpose only. Only an experience DBA must use this type of connection in business/production environment.

Security Aspect of connecting as 'sys'

A screenshot of a terminal window with a red title bar. The title bar text is 'oracle@localhost:/u01/app/oracle/product/12.1.0/dbhome_1/network/admin'. The terminal shows the contents of the 'sqlnet.ora' file. The text is as follows:

```
File Edit View Search Terminal Help
# sqlnet.ora Network Configuration File: /u01/app/oracle/product/12.1.0/dbhome_1
/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)
SQLNET.AUTHENTICATION_SERVICES=(none)
```

Figure 2.1 – Sqlnet.ora file.

It is a must to authenticate the sys user using DB authentication or using a different authentication technique.

```
[oracle@localhost admin]$ echo "SQLNET.AUTHENTICATION_SERVICES=NONE" >>  
$ORACLE_HOME/network/admin/sqlnet.ora
```

Above command will add the "SQLNET.AUTHENTICATION_SERVICES" parameter to the sqlnet.ora file and will force the user to enter the password for the sys user as the value 'none' will disable any other authentication techniques except database authentication.

Ref

12.2

[Parameters for the sqlnet.ora File \(oracle.com\)](#)

19.3

[Parameters for the sqlnet.ora File \(oracle.com\)](#)

21c

[Parameters for sqlnet.ora Files \(oracle.com\)](#)

Getting familiar with Oracle Database Instance. Activity 2 continues.

Define Oracle Instance.

Theoretically, Oracle database instance is collection of Memory Structures and Background process. Along the Memory structures known as shared or system global Area (SGA).

Instance = MMS (SGA) + BGP

Practically, Instance is equal to Oracle database as without the instance a database cannot be up and running.

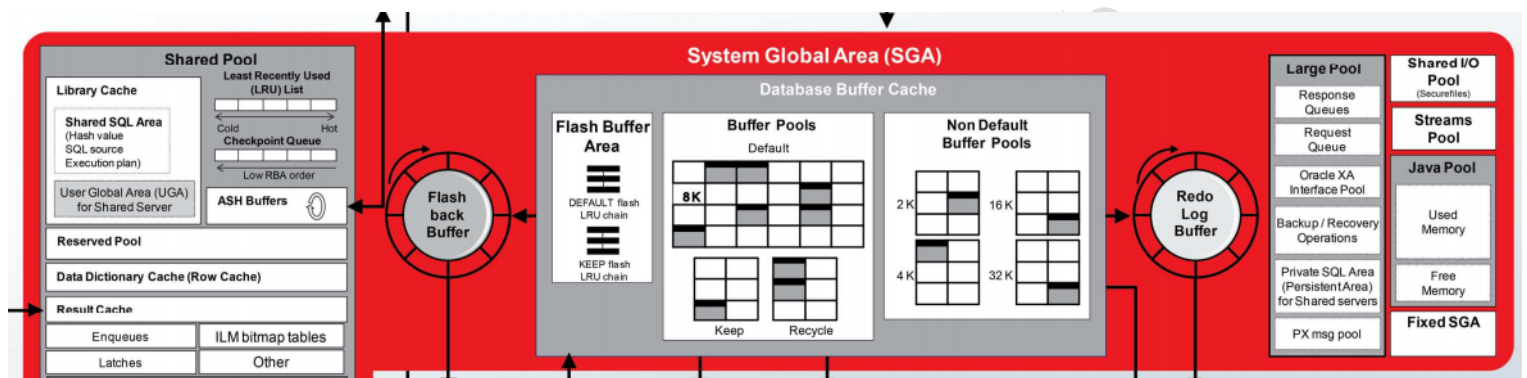


Figure 2.2 – Oracle Database Instance

Ref

http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/12c/r1/poster/OUTPUT_poster/poster.html

<https://www.oracle.com/webfolder/technetwork/tutorials/architecture-diagrams/19/pdf/db-19c-architecture.pdf>

<https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/background-processes.html#GUID-86184690-5531-405F-AA05-BB935F57B76D>

https://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/wls/12c/12c_poster/poster.html#

As in figure 2.2, the main memory structures are

Buffer pool / Data buffer Cache – which holds the data from data files.

Shared I/O pool – which hold the execution plans and data dictionary cache.

Redo Log Buffer – which holds the changes made to the records in the buffer pool.

The key back ground process are

CKPT - Checkpoint – synchronizes the control files and datafile

DBWr0 - Database Writer – writes data from data buffer cache to database files.

LGWR - Log Writer – writes data from redo log buffer to redo log files.

SMON - System Monitor - Performs critical tasks such as instance recovery and dead transaction recovery

PMON - Process Monitor - Scans for dead processes and coordinates cleanup

Let's take look at instance in action.

SQL> startup nomount

ORACLE instance started.

Total System Global Area 1560281088 bytes

Fixed Size 2924784 bytes

Variable Size 1006636816 bytes

Database Buffers 536870912 bytes

Redo Buffers 13848576 bytes

SQL> select status from v\$instance;

STATUS

STARTED

Startup nomount command will bring only the instance up and will establish the memory structures and background process in Virtual Memory. Figure 2.2 shows the graphical representation of this.

At this moment only the logical structures of the Database are available.

Let's mount the database

SQL> alter database mount;

Database altered.

SQL> select status from v\$instance;

STATUS

MOUNTED

At this stage Oracle server links the database with control file in the physical structure.

And the control file gets opened. It contains all the names, paths of the redo logs and the datafiles which are the rest of the database files and also belongs to physical structure.

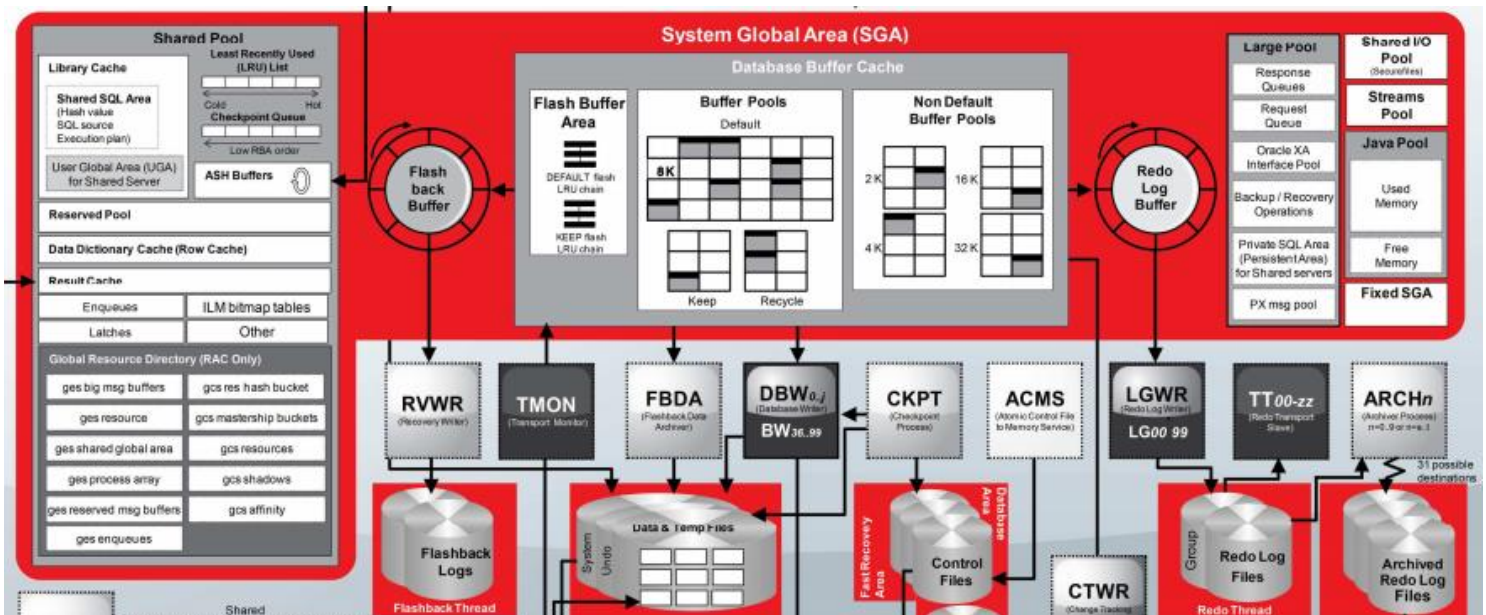


Figure 2.3 Oracle Database Logical and Physical Structures.

Let's open the database

SQL> alter database open;

Database altered.

SQL> select status from v\$instance;

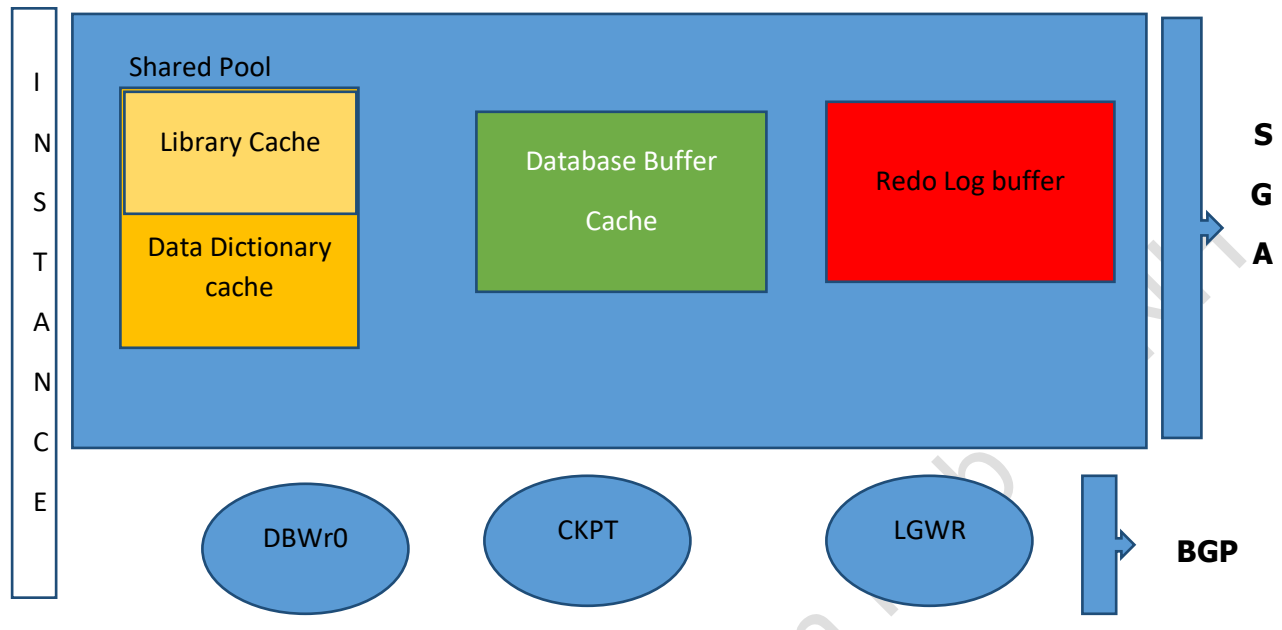
STATUS

OPEN

At this level oracle server will open all the log files and database files as described in the control file.

Figure 2.4 shows a simplified Oracle Database Logical and Physical Structure.

Instance (Logical Structure)



Database (Physical Structure)

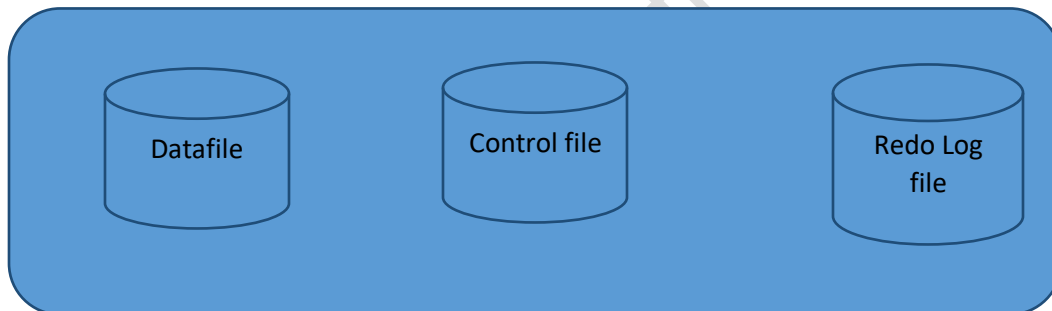


Figure 2.4 Oracle Database Logical and Physical Structures Simplified.

Oracle is manageable by design; it gives more control over instance management.

Let's try out few more commands

To shutdown the database

SQL> shutdown immediate;

Database closed.

Database dismounted.

ORACLE instance shut down.

Restart a database

SQL> startup

ORACLE instance started.

Total System Global Area 1560281088 bytes

Fixed Size 2924784 bytes

Variable Size 1006636816 bytes

Database Buffers 536870912 bytes

Redo Buffers 13848576 bytes

Database mounted.

Database opened.

SQL> startup force

ORACLE instance started.

Total System Global Area 1560281088 bytes

Fixed Size 2924784 bytes

Variable Size 1006636816 bytes

Database Buffers 536870912 bytes

Redo Buffers 13848576 bytes

Database mounted.

Database opened.

There is a risk here as the shutdown process is not visible once the startup force command issued. It does an abort shutdown.

SQL> startup force nomount

ORACLE instance started.

Total System Global Area 1560281088 bytes

Fixed Size 2924784 bytes

Variable Size **1006636816 bytes**
Database Buffers **536870912 bytes**
Redo Buffers **13848576 bytes**

ORACLE instance started.

Total System Global Area 1560281088 bytes

Fixed Size **2924784 bytes**
Variable Size **1006636816 bytes**
Database Buffers **536870912 bytes**
Redo Buffers **13848576 bytes**

Database mounted.

Oracle database shutdown process

<https://docs.oracle.com/database/121/ADMIN/start.htm#ADMIN11156>

When Oracle Restart is not in use, you can shut down a database instance with SQL*Plus by connecting as SYSOPER, SYSDBA, SYSBACKUP, or SYSDG and issuing the SHUTDOWN command. If your database is being managed by Oracle Restart, the recommended way to shut down the database is with the srvctl stop database command.

Control is not returned to the session that initiates a database shutdown until shutdown is complete. Users who attempt connections while a shutdown is in progress receive a message like the following:

```
ORA-01090: shutdown in progress - connection is not permitted
```

To shut down a database in normal situations, use the **SHUTDOWN** command with the **NORMAL** clause:

```
SHUTDOWN NORMAL
```

The **NORMAL** clause is optional, because this is the default shutdown method if no clause is provided.

Normal database shutdown proceeds with the following conditions:

- No new connections are allowed after the statement is issued.
- Before the database is shut down, the database waits for all currently connected users to disconnect from the database.

The next startup of the database will not require any instance recovery procedures.

Use immediate database shutdown only in the following situations:

- To initiate an automated and unattended backup
- When a power shutdown is going to occur soon
- When the database or one of its applications is functioning irregularly and you cannot contact users to ask them to log off or they are unable to log off

To shut down a database immediately, use the **SHUTDOWN** command with the **IMMEDIATE** clause:

SHUTDOWN IMMEDIATE

Immediate database shutdown proceeds with the following conditions:

- No new connections are allowed, nor are new transactions allowed to be started, after the statement is issued.
- Any uncommitted transactions are rolled back. (If long uncommitted transactions exist, this method of shutdown might not complete quickly, despite its name.)
- Oracle Database does not wait for users currently connected to the database to disconnect. The database implicitly rolls back active transactions and disconnects all connected users.

The next startup of the database will not require any instance recovery procedures.

When you want to perform a planned shutdown of an instance while allowing active transactions to complete first, use the **SHUTDOWN** command with the **TRANSACTIONAL** clause:

SHUTDOWN TRANSACTIONAL

Transactional database shutdown proceeds with the following conditions:

- No new connections are allowed, nor are new transactions allowed to be started, after the statement is issued.
- After all transactions have completed, any client still connected to the instance is disconnected.
- At this point, the instance shuts down just as it would when a **SHUTDOWN IMMEDIATE** statement is submitted.

The next startup of the database will not require any instance recovery procedures.

A transactional shutdown prevents clients from losing work, and at the same time, does not require all users to log off.

You can shut down a database instantaneously by aborting the database instance. If possible, perform this type of shutdown *only* in the following situations:

The database or one of its applications is functioning irregularly *and* none of the other types of shutdown works.

- You need to shut down the database instantaneously (for example, if you know a power shutdown is going to occur in one minute).
- You experience problems when starting a database instance.

When you must do a database shutdown by aborting transactions and user connections, issue the **SHUTDOWN** command with the **ABORT** clause:

SHUTDOWN ABORT

An aborted database shutdown proceeds with the following conditions:

- No new connections are allowed, nor are new transactions allowed to be started, after the statement is issued.
- Current client SQL statements being processed by Oracle Database are immediately terminated.
- Uncommitted transactions are not rolled back.
- Oracle Database does not wait for users currently connected to the database to disconnect. The database implicitly disconnects all connected users.

The next startup of the database *will* require instance recovery procedures.

Shutdown Timeout

Shutdown modes that wait for users to disconnect or for transactions to complete have a limit on the amount of time that they wait.

If all events blocking the shutdown do not occur within one hour, the shutdown operation aborts with the following message: ORA-01013: user requested cancel of current operation. This message is also displayed if you interrupt the shutdown process, for example by pressing CTRL-C. Oracle recommends that you do not attempt to interrupt an instance shutdown. Instead, allow the shutdown process to complete, and then restart the instance.

After ORA-01013 occurs, you must consider the instance to be in an unpredictable state. You must therefore continue the shutdown process by resubmitting a **SHUTDOWN** command. If

subsequent SHUTDOWN commands continue to fail, you must submit a SHUTDOWN ABORT command to bring down the instance. You can then restart the instance.

Instance recovery and checkpoints will be covered in detail in Architecture lessons.

Shutdown Modes	A	I	T	N
Allows new connections	No	No	No	No
Waits until current sessions end	No	No	No	Yes
Waits until current transactions end	No	No	Yes	Yes
Forces a checkpoint and closes files	No	Yes	Yes	Yes

Shutdown modes:

- A = ABORT
- I = IMMEDIATE
- T = TRANSACTIONAL
- N = NORMAL

In this activity the competency was built on

- Database authentication
- Security aspect of database authentication
- Getting familiar with Oracle Database instance and its components
- The database startup process
- Database shutdown process.

Next:

Understanding The Parameter File and Basic Trouble Shooting with Alert Log - CP 3