Database Systems II B: DBMS-Implementation

Chapter 4: Basic Oracle Architecture and Administration

Prof. Dr. Stefan Brass
Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2021/22

http://www.informatik.uni-halle.de/~brass/dbi21/

Objectives

After completing this chapter, you should be able to:

- enumerate processes, files, memory structures of the Oracle architecture.
- explain why delayed writing of changed database blocks is a good idea, and how the logfile protects changes.
- start and stop the Oracle server,
  enumerate different system states of the server.

Contents

Oracle Files: Data Files (1)

- Oracle normally stores table data in standard operating system files.

  Windows e.g.: `C:\Oracle\ORADATA\orcl\System01.dbf`

  UNIX e.g.: `/ora8/oradata/ifidb/system01.dbf`

- Alternatively, Oracle can store the data on raw devices (direct disk access, not via the OS): Better performance, but more complicated administration.

- The files can only be processed by Oracle (no standard format, the format is also not documented).

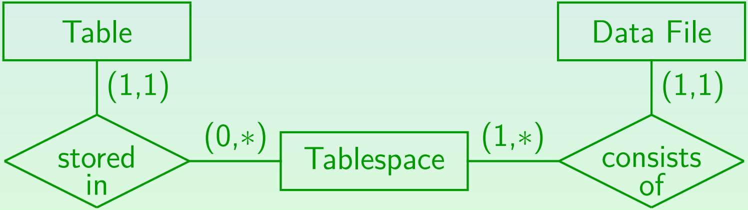
  Every DBMS vendor tries to beat the other vendors in performance benchmarks. Therefore, each vendor uses its own data structures.

Oracle Files: Data Files (2)

- Oracle does not use one file per table or per user.

  Any number of tables, indexes, etc. can be stored in the same file. Simplest case: Entire DB in a single file.

- The relationship between tables and data files is many to many (via tablespaces):



Oracle Files: Data Files (3)

- The data is not encrypted:
  - Persons who can access the data files can circumvent the Oracle access control.
  - OS access rights must be used so that only the DBA can access the data files.
- Data files can be autoextensible or have fixed size.

  In order to avoid fragmentation, data files are normally made large when they are created (the DBA can specify any size). Then Oracle manages the free space within them. If Oracle should ever run out of space, it can request more space from the operating system (make the file bigger) if the file was declared as autoextensible.

Oracle Files: Data Files (4)

- The data dictionary view DBA_DATA_FILES lists all files for storing table data. Columns are:
  - FILE_NAME: Filename with path.
  - FILE_ID: Numeric file identification.
  - TABLESPACE_NAME: Logical collection of data files.
  - BYTES, BLOCKS: Current file size.
  - STATUS: AVAILABLE or INVALID (not in use).
  - RELATIVE_FNO: File ID used in ROWIDs.
  - AUTOEXTENSIBLE: Oracle can make the file larger.
  - MAXBYTES, MAXBLOCKS: Limit for autoextension.
  - INCREMENT_BY: Step size for autoextension.

Oracle Files: Tempfiles (1)

- When large tables/query results must be sorted, Oracle needs temporary space on the disk.

  Small sorts are done in main memory.

- This is allocated in temporary segments.

  Temporary segments are also used for temporary tables.

- At the beginning, temporary segments were allocated in normal tablespaces, then Oracle introduced temporary tablespaces, and finally (8i) temporary datafiles (tempfiles) for temporary tablespaces.

Oracle Files: Tempfiles (2)

- Each variant of temporary data management is more efficient than the previous one, but all old variants are still supported.
- No backup copies of tempfiles are ever needed.

  Remember that the correctness of all information in this course is not guaranteed. You cannot sue me or my university for errors.
- Tempfiles are listed in
  - `DBA_TEMP_FILES`
  - `V$TEMPFILE`.
- Tempfiles typically have the extension "`.tmp`".

Oracle Files: Control Files (1)

- When Oracle starts, it reads the names and locations of the datafiles from a “control file”.

  Windows e.g.: `C:\Oracle\ORADATA\orcl\Control01.ctl`.

  UNIX e.g.: `/ora8/oradata/ifidb/control01.ctl`.

- The control file contains also backup and recovery information.
- For safety reasons, there should normally be more than one control file (on different disks).

  If all copies of the controlfile are lost, the DBA is in big trouble.

Oracle Files: Control Files (2)

- V$CONTROLFILE: List of control files. Columns are:
  - STATUS: Normally null. Can be INVALID.
  - NAME: Path and name of the control file.
- Information from the control file is shown in, e.g.:
  - V$DATAFILE,
  - V$DATABASE,
  - V$LOG,
  - V$LOG_HISTORY.

Oracle Files: Redo Log (1)

- When data file blocks are updated, they are not immediately written back to the disk.

  A block is the unit of exchange between disk and main memory: E.g., the system reads and writes always 8 KB. For performance reasons, disk blocks from the data files are kept for some time in a main memory buffer, even when they were modified (delayed/lazy writing).

- However, all changes to the data files are logged in the redo log files.

  It is faster to write only the new/modified data to a sequential log file than to write all parts of the data files that are affected by the change. Sooner or later the data file must be written, but this can happen in the background when there is time. It is possible that the same block is changed several times before to is written to the disk.

Oracle Files: Redo Log (2)

- The redo log files are needed for transaction processing (Recovery after a system crash.)

  Windows e.g.: `C:\Oracle\ORADATA\orcl\Redo01.log`.

  UNIX e.g. `/ora8/oradata/ifidb/redo01.log`

- Since the redo log files are so important for recovery, one usually has two copies of every log file.

  Of course, they should be on different disks.

- The log files that are copies of each other are called a log file group.

  Oracle automatically writes the same information to all files in a log file group. In contrast, Oracle does not manage copies of data files.

Oracle Files: Redo Log (3)

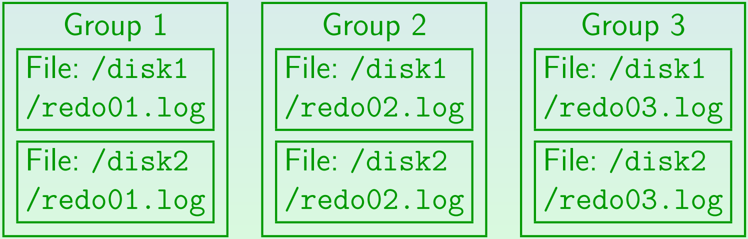
- When the changes that are documented in the log file are reflected in the data files it can be overwritten with new changes.

  However, if one wants to be protected against loss of data files (e.g., because of a disk failure), one needs to keep all redo log files since the last backup of the data files. This is done by copying redo log files to an "archive destination" (a tape or a slower disk) before they are overwritten. If Oracle is put into "`ARCHIVELOG`" mode (this is not the default), it automatically ensures that the redo log files are archived before they are overwritten. In order to distinguish the main redo log files from their archived copies, they are called the "online log". More information will be given later (Backup&Recovery).

Oracle Files: Redo Log (4)

- Redo log files are reused in a cyclic way, e.g. the output first goes to group 1, then to group 2, then to group 3, and then again to group 1.

Every Oracle instance needs at least two log files.



Group 1

File: /disk1 /redo01.log

File: /disk2 /redo01.log

Group 2

File: /disk1 /redo02.log

File: /disk2 /redo02.log

Group 3

File: /disk1 /redo03.log

File: /disk2 /redo03.log

Oracle Files: Redo Log (5)

- V$LOGFILE: List of all log files.
  - GROUP#: Logfiles with the same group number are copies of each other (for safety reasons).
  - STATUS: Usually null.

    It can be also be INVALID (file is inaccessible), STALE (file contens are incomplete), DELETED (file is no longer used).
  - MEMBER: File name of the log file.
- V$LOG contains information about the log file groups and their contents.

  Columns: GROUP#, THREAD#, SEQUENCE#, BYTES, MEMBERS, ARCHIVED, STATUS, FIRST_CHANGE#, FIRST_TIME.

Oracle Files: Parameters (1)

- When Oracle is started, it reads an initialization parameter file.

  Windows e.g.: `C:\Oracle\Admin\orcl\pfile\init.ora`

  UNIX e.g.: `/ora8/product/8.1.6/admin/ifidb/pfile/initifi.ora`

  Note that it might point to/include another file (`PFILE=...`).

- It contains the settings of important tuning parameters, as well as the location of the control files.
- Traditionally, the initialization parameter file was a standard ASCII file that could be viewed and modified with any text editor.

Oracle Files: Parameters (2)

- Beginning with Oracle 9i, “Server Parameter Files” are used that are binary files and can be changed only by Oracle.

  Traditional text files are still supported and it is possible to map in both directions with `CREATE SPFILE` and `CREATE PFILE`. Parameters are changed with `ALTER SYSTEM` (some also with `ALTER SESSION`). The advantage of server parameter files is that Oracle can automatically modify them for `ALTER SYSTEM` commands.

- Some parameters can be set only when the DBMS starts up, other parameters can be modified while the DBMS is running, and some parameters can be set separately for each user session.

Oracle Files: Parameters (3)

- V$PARAMETER: Settings of the initialization parameters that are in effect for the current session.

  Columns: NUM, NAME, TYPE, VALUE, ISDEFAULT, ISSES_MODIFIABLE (i.e. this parameter can be set separately for each session), ISSYS_MODIFIABLE (i.e. this parameter can be modified while the DBMS is running), ISMODIFIED, ISADJUSTED, DESCRIPTION, UPDATE_COMMENT. There is also a view V$PARAMTER2 that displays list-valued parameters differently (one row per list element), and V$SYSTEM_PARAMETER/...2 that contain the global values (inherited by each session when it starts).

- E.g. location of the control files:

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

- SQL*Plus has a command SHOW PARAMETER *X*.

Oracle Files: ALERT File (1)

- The alert file of the DB contains information about:
  - each time the server is started or stopped,

    At startup, the size of the shared memory areas and the started background processes are shown.
  - important administrative operations,

    For instance, adding files to the database.
  - errors.

    E.g., the archive log disk is full, therefore soon no more updates will be possible (when the online log files are used up).

Oracle Files: ALERT File (2)

- The DBA should carefully monitor the alert file.

  Windows, e.g.: `C:\Oracle\Admin\orcl\Bdump\orclALRT.log`

  UNIX, e.g.: `/ora8/product/8.1.6/admin/ifidb/bdump/alert_ifi.log`

- Some DBAs always have a window open that shows the last lines of the alert file.

  Of course, the Oracle Enterprise Manager also has a window that shows the current status of the database. One can also specify that when certain events happen, an email or SMS is automatically sent.

- In addition, each Oracle process has a trace file.

  Server processes only if the parameter `SQL_TRACE` is set to `TRUE`.

Contents

1 Oracle Files

2 Oracle Architecture

3 Startup and Shutdown

Oracle Architecture (1)

- Clients (such as SQL*Plus or an application program written in Pro*C) connect over the network to the DB server.
- The server part of Oracle consists of several processes and a shared memory area (SGA = "System Global Area" or "Shared Global Area").
- This server part is called an Oracle instance.

  Normally, an instance is synonymous to database. But with "Oracle real application clusters" it is also possible that several instances (using different CPUs) access the same database. Of course, it is possible to have several Oracle instances running on the same machine (managing different databases).

Oracle Architecture (2)

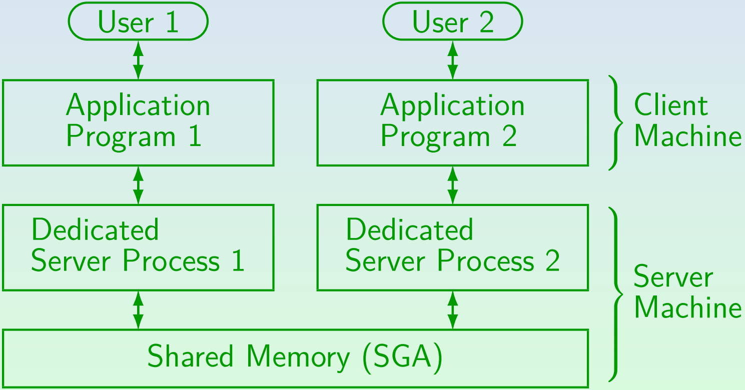
- Oracle supports two architectural variants:
  - Dedicated server architecture: One server process per client (classical/old architecture).

    The process is started when the client connects to Oracle and terminated when it logs off. Problem: The process is idle most of the time, but still binds resources (memory).
  - Multithreaded server architecture (MTS): There is a pool of server processes and a dispatcher which sends client requests to one of the servers.

    This is advantageous when a large number of concurrent clients must be served.

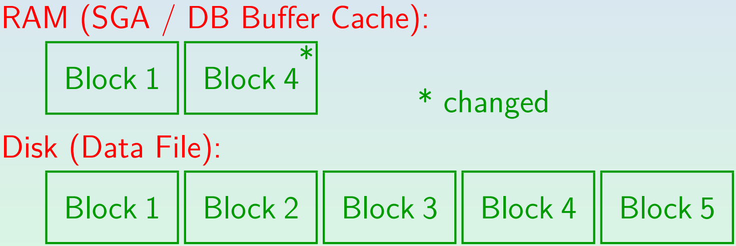
Oracle Architecture (3)

Dedicated Server Architecture:



Oracle Architecture (4)

- Part of the SGA is the DB buffer cache. It stores recently accessed DB blocks (containing e.g. table data).



RAM (SGA / DB Buffer Cache):

Disk (Data File):

- Accesses to RAM are much faster than accesses to disk, but the RAM is usually smaller (and volatile).

Oracle Architecture (5)

- When a server process needs a block from a datafile, it first checks whether the block is already in the DB buffer cache.
- If the block is already in the buffer cache, the server process can directly access the data there.

  Unless it is locked, of course.
- If not, the server process allocates a free buffer and reads the block into that buffer.

  The block then remains some time in the buffer, in case it is needed again.

  Buffering is explained in more detail in Chapter 6.

Oracle Architecture (6)

- Even if the server process changes the block in the buffer, it does not write the block back to the disk.
- The "DB writer" background processes (DBW0, DBW1, . . .) save changed blocks from the DB buffer cache periodically back to the disk.

  Because of the delayed writing, it might be that the block is changed many times before it is finally written back.
- Oracle uses a set of 5 or more background processes (besides the server processes).

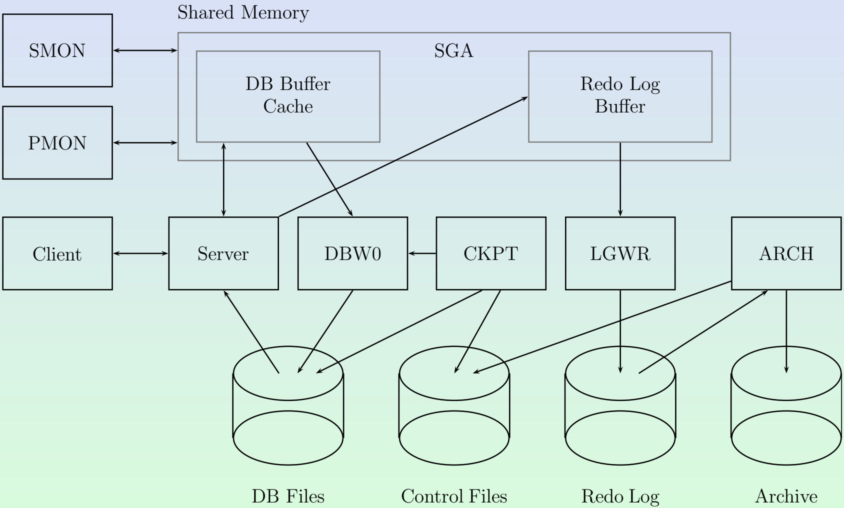
Oracle Architecture (7)

- Another part of the SGA is the Redo Log Buffer. It stores a transcript of all changes to DB blocks.

  This information is needed e.g. if the system should crash before the DB Writer saved a block back to disk. The Log entries are written to disk at commit time (or earlier).

- The log writer process (LGWR) saves the entries from the log buffer to the current redo log file.

Oracle Architecture (8)



Oracle Architecture (9)

More Background Processes:

- Checkpoint Process (`CKPT`)

  CKPT ensures that from time to time all changed blocks are written back to disk so that older log files are only needed in case of disk failures, but not for "normal" system crashes. It calls the DB writer process and updates the controlfiles. A checkpoint is set (minimally) every time a log file becomes full. One can configure more frequent checkpoints (`init.ora`).

- Archiver Process (`ARCH`)

  ARCH writes full redo log files to tape storage or another archive location (if the DB runs in ARCHIVELOG mode). If the data files should be damaged, all redo log files generated since the last backup are needed.

Oracle Architecture (10)

Oracle Background Processes, Continued:

- System Monitor Process (`SMON`)

  SMON performs recovery after a system crash and does some clean-up tasks regularly. E.g. it merges contiguous free extents etc.

- Process Monitor Process (`PMON`)

  PMON performs clean-up tasks when a user process fails (e.g. remove its locks).

Oracle Instance:

- An Oracle instance consists of an SGA (system global area) and a set of background processes.

Oracle Architecture (11)

Main Networking Process:

- Listener

  The network listener process accepts connections from clients and creates dedicated server processes. It can also manage pre-spawned server processes.

Processes for Multithreaded Servers:

- Dispatcher Processes (`Dnnn`)

  In the multithreaded configuration, these processes distribute client requests over shared DB servers.

- Shared Server Processes (`Snnn`)

  DB Server for the multithreaded configuration.

Oracle Architecture (12)

Processes for Distributed Databases:

- Recoverer Process (RECO)

  This process connects to other nodes in a distributed DB to resolve in-doubt transactions (no final COMMIT/ABORT).

- Job Queue Processes (SNPn)

  These processes automatically update table snapshots in a distributed database.

Processes for Oracle Parallel Server:

- Lock Process (LCK0)

  Manages locks between different Oracle instances (each instance has its own SGA and background processes).

Oracle Architecture (13)

Memory Structures:

- Besides the DB buffer cache and the redo log buffer, the SGA (System Global Area) contains also the “Shared Pool”.
- The shared pool contains e.g.
  - the “row cache” (for data dictionary entries).

    It seems that caching single rows is more efficient for the data dictionary than caching entire blocks.
  - the library cache.

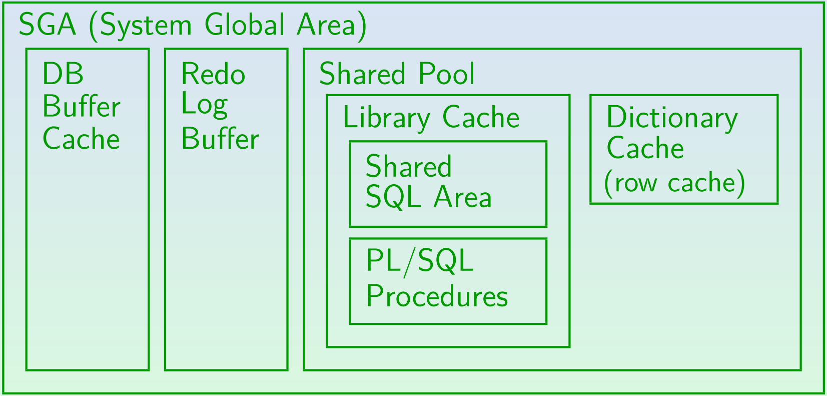
Oracle Architecture (14)

Memory Structures, continued:

- The library cache contains e.g.
  - the Shared SQL Area (here recently executed SQL statements together with the query evaluation plans are stored).

    This is a cache for query evaluation plans: If the same SQL statement (e.g., from an application program) is executed again and again (possibly with different parameter values), the (relatively expensive) query optimization does not have to be repeated.
  - Stored procedures/packages in compiled form.

Oracle Architecture (15)



SGA (System Global Area)

DB Buffer Cache

Redo Log Buffer

Shared Pool

Library Cache

Shared SQL Area

PL/SQL Procedures

Dictionary Cache (row cache)

Oracle Architecture (16)

Program Global Area (PGA):

- The PGA is memory that is allocated inside the dedicated server process (i.e. not shared).
- It contains e.g.
  - Stack area (session-specific variables, arrays, ...)
  - Private SQL areas (bind information, runtime buffers, etc.)
- The private SQL areas contain also the sort areas.

  Sorting can run faster with more memory. Only the retained portion of the sort area is part of the private SQL area.

Oracle Architecture (17)

Remark about Multithreaded Server:

- In the multithreaded server configuration, the private SQL areas are allocated in the SGA (in the shared pool).

  That means the size of the SGA must be increased when changing from the dedicated server configuration to the multithreaded server configuartion.

Oracle Architecture (18)

Related Information in the Data Dictionary:

- V$PROCESS: List of the Oracle processes.
- V$BGPROCESS: Description of background processes.
- V$BUFFER_POOL: Size of DB buffer cache.

  Oracle can have different “buffer pools” with different replacement strategies. “BUFFERS” is the number of buffer frames. The buffer pools can be segmented into multiple sets for multiprocessor systems. Buffering is explained in more detail in the next chapter.

Oracle Architecture (19)

Related Information in the Data Dictionary, continued:

- V$RESOURCE_LIMIT: Sizes of some arrays in the SGA.

  E.g. the SGA contains an array for the currently active transactions. V$RESOURCE_LIMIT also reports the use of these resources, e.g. how many transactions are currently active or were ever active concurrently (since instance startup).

- V$SGA: Size of the components of the SGA.

  E.g. it contains the size of the DB buffer pool in bytes.

- V$SGASTAT: Detailed information about the size of memory structures, listing components of the shared pool individually.

Initialization Parameters

- DB_BLOCK_SIZE: Size (in bytes) of a single DB block.
- DB_BLOCK_BUFFERS: Number of buffer frames.

  So the total memory needed is DB_BLOCK_BUFFERS * DB_BLOCK_SIZE.
- LOG_BUFFER: Size of the redo log buffer (in bytes).
- SHARED_POOL_SIZE: Size (in bytes) of the cache for data dictionary rows, query evaluation plans, etc.
- DB_FILE_MULTIBLOCK_READ_COUNT: Number of blocks that are read in one OS call during full table scans.
- There are more than 200 such parameters.

Contents

1 Oracle Files

2 Oracle Architecture

3 Startup and Shutdown

Controlling Oracle (1)

- The DB server can be running (i.e. executing queries and updates) or not running (shut down).

  There are also several intermediate/restricted states, which are necessary for DB creation or recovery, or certain administrative operations.

- One of the tasks of the DBA is to control the availability of the database.

- For adminitrative operations (such as DB startup), the DBA should log into Oracle as follows:

```
UNIX> sqlplus /nolog
SQL> CONNECT SYS AS SYSDBA
```

Controlling Oracle (2)

- "/nolog" means that no connection to a database is opened (e.g. when the database is not yet started).

  This might not be necessary, since when one logs in "AS SYSDBA", one gets only a warning when the database is not yet started.

- "AS SYSDBA" gives special administrative rights, e.g. the right to startup the database or shut it down.

  "SYSDBA" is a system privilege in Oracle, but it is can also be viewed as a special type of connection to the database. In other words, it is a privilege that must be explicitly activated. While SYS can only log in AS SYSDBA, other users who were granted the SYSDBA right can choose whether they log in AS SYSDBA or as a normal user. There is also a weaker right called SYSOPER, which permits to start up or shut down the database, but does not permit to view all data in the database.

Controlling Oracle (3)

- As long as the DBMS does not run, the passwords for SYS etc. stored in the DB cannot be accessed.
- There are two possibilities to authenticate DBAs:
  - OS Authentication: Members of an operating system group called dba may start/stop the DB.

    E.g. the OS user who owns the Oracle programs and files (usually oracle). On other systems, only the OS administrator (root) may start/stop the DB.
  - Password File: There is a password file in addition to the passwords stored in the database.

    It contains the passwords of all users who have been granted the SYSDBA or SYSOPER system privileges.

Controlling Oracle (4)

- The password file is e.g. stored in (OS dependent):

```
C:\orawin95\database\pwdorcl.ora
$ORACLE_HOME/dbs/orapw$ORACLE_SID
```

- The program "orapwd" creates the password file:

```
orapwd FILE=orapworcl PASSWORD=nina ENTRIES=5
```

- PASSWORD is the initial password for SYS.
- ENTRIES is the maximal number of users with SYSDBA or SYSOPER rights (i.e. with passwords in this file).

  The file has a fixed size. Whenever a user is granted SYSDBA or SYSOPER, an entry in the password file is created.

Controlling Oracle (5)

- If one forgot the password: Delete or rename the password file and recreate it.

  The DB must be restarted. Afterwords one can log in `AS SYSDBA` and change the password in the DB (the two `SYS` passwords do not agree).

- A password file is used if the initialization parameter `REMOTE_LOGIN_PASSWORD_FILE` is set to `EXCLUSIVE`.

  It is `NONE` by default which means OS authentication.

- On Windows, a DBA password is stored in the registry. `DBA_AUTHORIZATION` in `HKEY_LOCAL_MACHINE/SOFTWARE/ORACLE`.

  This is used for the automatic start and stop of the DBMS.

Controlling Oracle (6)

- All users who connect AS SYSDBA are mapped to SYS when they access the DB.

  It might be a surprise that one's own tables are not available when one works in this mode. Users who work AS SYSOPER are mapped to PUBLIC.

- The DBA usually works while logged into the server machine.

  Of course, he/she can use ssh to log into the server from somewhere else. It is possible to do administration remotely (with SQL*Plus running on the client), but normally the network connection is not very secure. Also, unless one uses the new server parameter files, a copy of the initialization parameter file must be available on the client.

Starting Oracle (1)

- Starting Oracle proceeds in three steps. Normally, they are automatically done one after the other, but it is possible to remain in an intermediate state:
  - The instance is started.

    The parameter file is read, the SGA is allocated, the background processes are started. In this state a new DB can be created.
  - The database is mounted.

    The control file is read, but the data files are not yet open. E.g. data files can be renamed in this state.
  - The database is open.

    The datafiles and log files are open, and the database is available for normal operations.

Starting Oracle (2)

- The commands for starting Oracle in one of the above states are:
  - `STARTUP NOMOUNT`: Only processes started.
  - `STARTUP MOUNT`: Only controlfiles open.
  - `STARTUP` or `STARTUP OPEN`: Full startup.
- It is possible to add the keyword `FORCE`, this e.g. kills processes remaining from a previous instance.
- `STARTUP RESTRICT` opens the database, but allows only users with the `RESTRICTED SESSION` privilege to connect (e.g. only DBAs).

Starting Oracle (3)

- One can explicitly specify the parameter file and/or the DB name:

  ```
  STARTUP OPEN ifi PFILE=initifi.ora
  ```

  The default PFILE is $ORACLE_HOME/dbs/init$ORACLE_SID.ora on UNIX, and %ORACLE_HOME%\database\initORCL.ora on Windows. However, in starting in Oracle 9i, the default is to use a server parameter file. If one uses a traditional text parameter file, one must specify PFILE=....

- A database in mounted state can be made available for users with

  ```
  ALTER DATABASE OPEN
  ```

Shutting Oracle Down (1)

- The shutdown proceeds in the same three steps as the startup (in inverse order):
  - The database is closed.

    The contents of the DB buffer cache and the redo log buffer is written to disk and the files are closed. The control files remain open.
  - The database is dismounted.

    The control files are closed.
  - The instance is shut down.

    The background processes are terminated, the SGA memory is given back to the operating system.

Shutting Oracle Down (2)

- There are four different shutdown modes:
  - SHUTDOWN NORMAL

    The shutdown waits until all users logged off from Oracle. No new users can log into Oracle.
  - SHUTDOWN TRANSACTIONAL

    The shutdown waits until all active transactions are finished. No new transactions can be started.
  - SHUTDOWN IMMEDIATE

    All active transactions are rolled back. Then all buffers are written and the shutdown proceeds normally.
  - SHUTDOWN ABORT

    The Oracle processes are killed. Recovery will be needed.

Killing Sessions (1)

- All currently active sessions (i.e. users that are logged in) are listed in `V$SESSION`. Columns are, e.g.:
  - `SADDR`: Memory address of session data in DBMS.
  - `SID`: Session identifier.
  - `SERIAL#`: Serial number of the session.

    The `SID` is reused for another session when one user logs out and the next logs in. The serial number is added to make it unique.
  - `USERNAME`: Oracle user name.
  - `OSUSER`: Operating system user name.
  - `PROGRAM`: Operating system program name.

Killing Sessions (2)

- Before the DBA shuts down the system or kills a session, he/she might check whether transactions are currently running.

  A transaction is started at the first `INSERT`/`UPDATE`/`DELETE` and ends with `ROLLBACK` or `COMMIT`. I.e. if the session should be killed, changes are rolled back (a user will have to enter data again).

- `V$TRANSACTION` lists the currently active transactions. Attributes are, e.g.,:

  - `SES_ADDR`: Memory address of the session data.
  - `START_TIME`: Time when the transaction started.

Killing Sessions (3)

- Exercise: Write a query to list the users who have currently active transactions.

- `V$TRANSACTION_ENQUEUE` lists the locks hold by transactions.

  One attribute is the `SID` of the session that holds the lock. `CTIME` is the time since the transaction holds the lock (in seconds). `BLOCK` is 1 if another transaction waits for the lock. `V$LOCK` lists all locks and lock requests (including system locks: The list is quite long).

- The DBA can kill a session with the command:

  ```
  ALTER SYSTEM KILL SESSION '8,5948'
  ```

  8 is the `SID` and 5948 is the `SERIAL#` of the session.

References

- Ramez Elmasri, Shamkant B. Navathe: Fundamentals of Database Systems, 3rd Ed. Chapter 17: “Database System Architectures and the System Catalog”, Chapter 10: “Examples of Relational Database Management Systems: Oracle and Microsoft Access”
- Michael J. Corey, Michael Abbey, Daniel J. Dechichio, Ian Abramson: Oracle8 Tuning. Osborne/ORACLE Press, 1998, ISBN 0-07-882390-0, 608 pages, ca. $44.99.
- Jason S. Couchman: Oracle8i Certified Professional: DBA Certification Exam Guide with CDROM. Osborne/ORACLE Press, ISBN 0-07-213060-1, ca. 1257 pages, ca. $99.99.
- Mark Gurry, Peter Corrigan: Oracle Performance Tuning, 2nd Edition (with disk). O’Reilly & Asociates, December 1996, ISBN 1565922379, 964 pages.
- Oracle 8i Concepts, Release 2 (8.1.6), Oracle Corporation, 1999, Part No. A76965-01.
- Oracle 8i Administrator’s Guide, Release 2 (8.1.6), Oracle Corporation, 1999, Part No. A76956-01.
- Oracle 8i Designing and Tuning for Performance, Release 2 (8.1.6), Oracle Corporation, 1999, Part No. A76992-01.