**Oracle Non-Container Architecture**

Oracle architecture is divided into 3 parts.

1. **Memory structure**
2. **Process structure**
3. **Storage structure**

**Memory structure:**

* Memory elements in oracle database are **SGA** and **PGA.**
* SGA is shared global area.
* PGA is programable global area or private global area.

**SGA:**

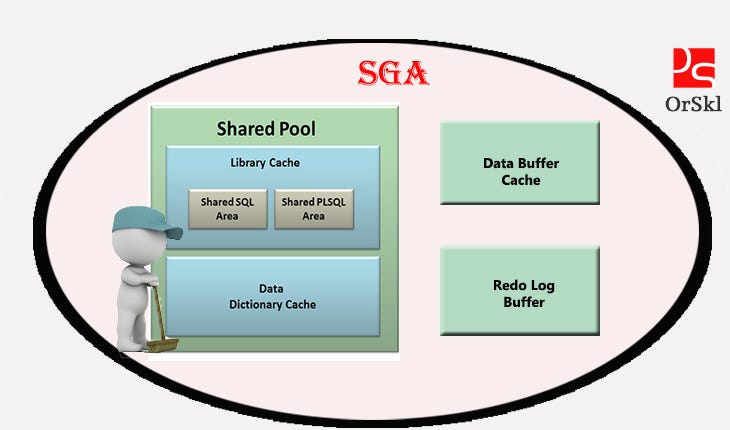
* SGA will starts when the instance is started.
* SGA is divide into 3 parts.

1. Shared pool. (again shared pool is divided into 2 parts)

* Library cache
* Data dictionary cache

1. Database buffer cache.
2. Redo log buffer cache.

* In addition to this we have **LARGE POOL** , **JAVA POOL** and **KEEP POOL**.



**Shared Pool:**

* It stores most recently executed SQL statements and most recently used data definations in the form of #values that information is stored in library cache.
* It consists of two key performance-related memory structures.
* Library cache & data dictionary cache
* Sized by parameter **SHARED\_POOL\_SIZE**
* If the SGA size is less shared\_pool will get filled.
* We can solve by flusing the shared pool.
* **ALTER SYSTEM FLUSH SHARED\_POOL**;
* All the data in shared pool will gone.

Library cache:

* Stores information about the most recent used SQL and PL/SQL statements.
* This is managed by a least recent used (LRU) algorithm.
* This consists of two areas.
* Shared SQL area
* Shared PL/SQL area

Data dictionary cache:

* This contains related metadata of each execution plan generated by optimizer and which stored in library cache.
* It includes about database files, tables, indexes, columns, users, privileges and other database objects.

**Database Buffer Cache:**

* It stores copies of data blocks that have retrived from the datafiles.
* Enables great performance when you obtain or update data.
* This is managed by a least recent used (LRU) algorithm.
* **DB\_BLOCK\_SIZE** determines primary block size.
* **DB\_CACHE\_SIZE** determines the size of **DBBC**.

**Redolog Buffer Cache:**

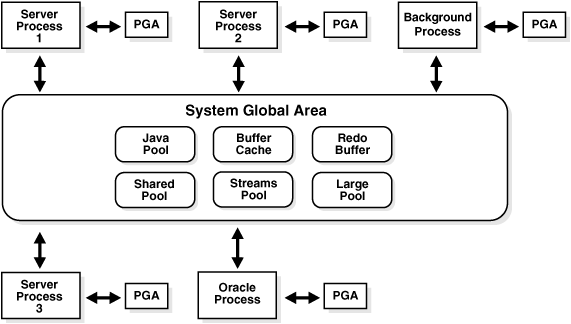
* Records all changes made to the database data blocks.
* Primary purpose is recovery.
* Changes recorded in rblc are called redo entries.
* Redo entries contain information to reconstruct or redo changes.
* Size defined by **LOG\_BUFFER**

**Keep Pool:**

* When the user is executing same query more times. That query blocks are stored in KEEP pool.

**PGA:**

* PGA will start when the server process starts.
* For every server process there is a separate PGA.
* If the size of PGA is 100mb and 4 sessions are there each session will get 25mb.
* If the PGA size is not enough then database will use temporary tablespace.



**Process structure**

* This is having 2 tpyes of background process.

1. **Mandatory background processes.**
2. **Non-mandatory background processes.**

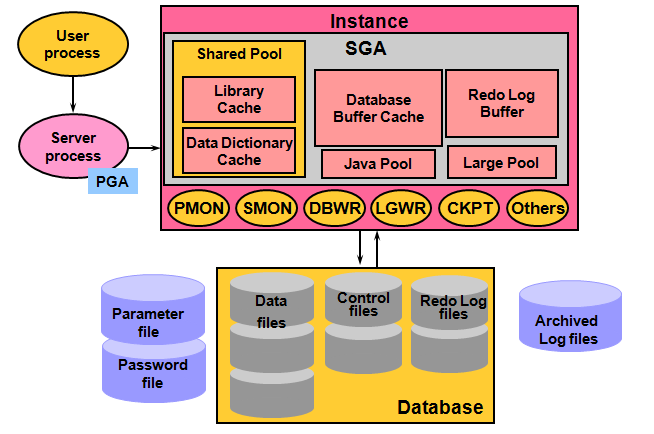
**Mandatory background process:**

* These processes are must required to run the oracle database.
* If any of this Background process killed instance will terminated.
* Instance = memory + background process.

1. PMON (**Process Monitor)**
2. SMON (**System Monitor)**
3. RECO (Recoverer)
4. DBWR (**Database Block Writer)**
5. LGWR (**Log Writer)**
6. CKPT (**Checkpoint Process)**

**Non-mandatory background processes:**

1. ARCn (Archiver process)
2. CTWR (Block change tracking)
3. MMON (Manageability monitor)
4. MMAN (Memory Manager)
5. JOB QUEUE PROCESS

****

**User / Client Process**:

* A program that requests interaction with the oracle database.
* Dose not interact directly with the oracle database.

**Server Process:**

* **When user process start then there is a dedicated server process associated with that user process it Connects to the oracle instance when a user establishes a session.**
* **A program that directly interacts with the oracle server.**
* **This can be dedicated or shared server.**
* **Reads required data blocks from datafiles on disk into the shared database buffers of SGA, if the blocks are not present in the SGA and returns results**

**Background Process:**

* **Started when an oracle instance is started.**

**Mandatory :**

**PMON**:

* If client dedicated server process broken, and transaction not committed then pmon will rollback that transaction and will clean dedicated server process.
* Monitor other background process. If any process failed then pmon will start that process.
* If there is any problem with redo log, then **PMON** will shutdown the database.
* In 11g PMON is responsible for registring db entries into dynamic listener.
* In 12c LREG is responsible for registring db entries into dynamic listener.

**SMON**:

* Whenever database got abnormal shoutdown **SMON** will perform instance recovery in next startup.
* It will roll farward committed transactions & roll back uncommitted transactions.

**RECO:**

* RECO process is used for the distributed database configurations.
* Whenever any distributed transactions failed. This process will note down the failed transactions and once the network is established between distributed databases then this process will sync the transactions.

**DBWR:**

* Initially modified blocks stores in database buffer cache. check point will continusly monitor dbbc and when dbbc got filled checkpoint will invoke DBWR and DATABASE BLOCK WRITER writes modified blocks into actual datafiles.
* Database writer process writes dirty buffers to disk under the following conditions.
* When the database issues checkpoint.
* When server process can’t find clean reusable buffer after checking a threshold number of buffers.
* Every 3 seconds.
* Dbwr uses Least Recently Used algorithm LRU.
* Oracle provides multiple database writer processes to share heavy data modification workload.
* You can have maximum of 20 dbwr processes.

**LGWR:**

* Lgwr writes the information available in REDOLOG BUFFER CACHE to actual redo log groups current redo.
* If current redo got filled 100% then log switch will happen and if archivelog mode is enabled. Then ARCHIVER process will generate physical archive.
* LGWR transfers content from RLBC to disk during the following conditions.
* Every 3 seconds.
* When redo log buffer filled up 1/3 of it’s size.
* When DBWR signal to LGWR to write.
* When commit occurs.
* Redolog buffer contains 1 mb of data.

**CKPT:**

* Ckpt is responsible for updating latest SCN to controlfile and datafiles headers.

**Non-Mandatory:**

**ARCn:**

* This process is non-mandatory. We can enable or disable this process.
* If this process is enabled whatever the data which is available in redolog that is converted as physical archive by ARCHIVER process.
* If archive log is enabled whenever data loss occur’s we can do recovery using archives.

**CTWR:**

* It is a change track writer.
* To increase the RMAN incremental backup speed we can enable **CTWR**.
* If this process is not enabled RMAN will read all the datafiles for modified blocks.

**MMON**:

* This process was interduced in oracle 10g.
* Mmon will take snap of the database for every 1 hour and stores in sysaux tablespace.

**MMAN:**

* This process was interduced in oracle 10g.

**ASMM (**AUTOMATICSHARED MEMORY MANAGEMENT**)**

* There is new parameter interduced from 10g that is **SGA\_TARGET**.
* Instead of setting sizes for all the SGA components we can directly set size for this parameter.
* Before 10g we have to set sizes for all SGA components. (shared pool, dbbc, rlbc, large pool, java pool, keep pool)
* **MMAN** will dynamically resizes the SGA components whenever required.

**JOB QUEUE PROCESS:**

* This process helps to run the scheduled jobs.
* By default from 11g its value is 1000.
* We can disable this by setting value to 0 and no job’s will work.

**Storage structure:**

* Storage elements like different tablespaces, Permanent tablespace, Temporary tablespace, Undo tablespace.

1. **system**
2. **sysaux**
3. **users**
4. **Temporary**
5. **Undo**

**How select statement works:**

* When client process executed Select \* from emp; Syntax will check in client process itself.
* If the syntax is correct client process passes the query to server process.
* Server process will check symantic check(privilages)
* It will check weather user has previlige on the table & Table & Rows is present or not.
* If everything is fine this query is converted into machine level language and oracle generate number of execution plans. (execution plan= best way to get output)
* Optimizer will pick the best execution plan.
* Optimizer’s are two types.
* Cost based optimizer
* Rule based optimizer
* Once the execution plan is selected by the optimizer that plan is stored in LIBRARY CACHE as a # value and server process will do parsing.
* Now server process will check weather query related data blocks are present in DBBC or not.
* If data blocks are not available in dbbc then server process will pick the data blocks from datafiles to DBBC. (This is called hard parsing or cache miss)
* from DBBC to server process and to client process.
* Finaly client will get result.

**Hard parsing**:

* If data blocks of query are not available in dbbc then server process will pick the data blocks from datafiles to DBBC This is called hard parsing or cache miss.

**Soft parsing**:

* If data blocks of query are available in DBBC This is called soft parsing or cache hit.

**How select statement works with sorting operations:**

* When client process executed Select \* from emp where no=1; Syntax will check in client process itself.
* If the syntax is correct client process passes the query to server process.
* Server process will check symantic check(privilages)
* It will check weather user has previlige on the table & Table & Rows is present or not.
* If everything is fine this query is converted into machine level language and oracle generate number of execution plans. (execution plan= best way to get output)
* Optimizer will pick the best execution plan.
* Optimizer’s are two types.
* Cost based optimizer
* Rule based optimizer
* Once the execution plan is selected by the optimizer that plan is stored in LIBRARY CACHE as a # value and server process will do parsing.
* Now server process will check weather query related data blocks are present in DBBC or not.
* If data blocks are not available in dbbc then server process will pick the data blocks from datafiles to DBBC. (This is called hard parsing or cache miss)
* from DBBC to PGA. Here for every server process there is associated PGA.
* In this PGA sorting operations will happen. If PGA size is not enough oracle will use’s temporary tablespace to perform sorting operation and gives output to the client.
* If temporary tablespace also not enough user will get unable to extend temp tablespace error.

**How undo update statement works:**

* When client process executed Update t1 set sal=300 where no=1; Syntax will check in client process itself.
* Then select will happen and check block is available in datafile or not.
* If the syntax is correct client process passes the query to server process.
* Server process will check symantic check(privilages)
* It will check weather user has previlige on the table & Table & Rows is present or not.
* If everything is fine this query is converted into machine level language and oracle generate number of execution plans. (execution plan= best way to get output)
* Optimizer will pick the best execution plan.
* Optimizer’s are two types.
* Cost based optimizer
* Rule based optimizer
* Once the execution plan is selected by the optimizer that plan is stored in LIBRARY CACHE as a # value and server process will do parsing.
* Now server process will check weather query related data blocks are present in DBBC or not.
* If data blocks are not available in dbbc then server process will pick the data blocks from datafiles to DBBC and also to UNDO tablespace.
* Old value is stored in undo tablespace based on undo\_retention policy.
* If user didn’t commit the transaction and undo\_retention crossed user will get ORA-1555 error.